8-2011

Experimental Testing of the Accuracy of Attitude Determination Solutions for a Spin-Stabilized Spacecraft

Keegan P. Ryan
Utah State University

EXPERIMENTAL TESTING OF THE ACCURACY OF ATTITUDE DETERMINATION SOLUTIONS FOR A SPIN-STABILIZED SPACECRAFT

by

Keegan P. Ryan

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved:

Dr. R. Rees Fullmer Major Professor	Dr. David Geller Committee Member
Dr. Stephen A. Whitmore Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2011

Abstract

Experimental Testing of the Accuracy of Attitude Determination Solutions for a Spin-Stabilized Spacecraft

by

Keegan P. Ryan, Master of Science

Utah State University, 2011

Major Professor: Dr. R. Rees Fullmer
Department: Mechanical and Aerospace Engineering

Spin-stabilized spacecraft generally rely on sun and three-axis magnetic field sensor measurements for attitude determination. This study experimentally determines the total accuracy of attitude determination solutions using modest quality sensors. This was accomplished by having a test spacecraft collect data during spinning motions. The data was then post-processed to find the attitude estimates, which were then compared to the experimentally measured attitude. This same approach will be used to test the accuracy of the attitude determination system of the DICE spacecraft to be built by SDL/USU.

(117 pages)

Acknowledgments

First, I would like to thank my major professor Dr. R. Rees Fullmer for all the help he has given me through the entire process of this thesis. It could not have been possible without his continual guidance and support. I would also like to thank Steve Wassom, Mitch Whiteley, Lynn Chidester, and many others from SDL, who were all so willing to help me solve the many problems I encountered with the hardware. I must also acknowledge my wife and kids for their love and encouragement that helped me more than they know.

Keegan Ryan

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Small Satellite Attitude Determination

Small satellites are currently used for many different types of space missions. These types of missions include military, science, space weather, communications, and education. As new technology continues to shrink electronic systems, there will be more possible missions for small satellites.

Whatever the mission may be, it is necessary that a spacecraft has knowledge of its attitude, or its orientation with respect to a celestial coordinate system. A spacecraft has an Attitude Determination System (ADS) to provide knowledge of the spacecraft's attitude. An ADS consists of sensors which make vector measurements and the software which processes the measurements to determine the attitude. The required accuracy of an ADS varies from mission to mission. Due to the unique requirements and constraints of each small satellite, it is not uncommon to develop a new ADS for many missions.

1.2 Dynamic Ionosphere CubSat Experiment (DICE)

The Space Dynamics Lab (SDL) is a not-for-profit University-affiliated research center owned by Utah State University. One of the current small satellite missions at SDL is the Dynamic Ionosphere Cubesat Experiment (DICE). The DICE program will launch two identical spinning spacecraft to measure plasma density and electric fields. The goal of the DICE program is to map the geomagnetic storm-enhanced density plasma bulge and plume formations in the Earth's ionosphere. The two CubeSats each use a sun sensor and a magnetometer in their ADS.

SDL has also developed the Nano-Satellite Operation Verification and Assessment (NOVA) test facility. The NOVA test facility is used to experimentally validate various

systems on small satellites. The capabilities of the NOVA test facility are: mass properties testing, magnetic field generation, solar cell testing, and ADS testing. Magnetic field generation and ADS testing will be the only capabilities of the NOVA lab that will be further explored. The NOVA test facility will be used to validate the DICE spacecraft.

1.3 Objectives

The overall objective of the this thesis is to experimentally determine the attitude determination accuracy of a generic spin stabilized sun and magnetometer-based ADS with modest quality sensors in the NOVA lab. This test system will validate the test cell and test procedures, plus provide a baseline for later attitude tests using the flight DICE spacecraft. This objective will be accomplished by completing the following:

- Design a single degree of freedom attitude determination test that will be conducted in the NOVA lab.
- Determine methodologies to calibrate three-axis magnetometers that are used in attitude determination systems.
- Determine methodologies to conduct the attitude determination tests.
- Determine methodologies to calculate the estimated attitude of the ADS and measure the corresponding error using the NOVA test cell.

1.4 Approach

1.4.1 Design a Single-Degree-of-Freedom Attitude Determination Test

For the testing of the ADS a ground simulator or mock satellite incorporating the ADS was developed. The mock satellite includes a magnetometer, a sun sensor, a light sensor, and a CPU. The sun sensor and magnetometer on the mock satellite had to be properly excited by external stimuli representing the Sun and the Earth's magnetic field to provide measurements. A laser, Helmholtz cage, and strobe were used to excite the sun sensor, magnetometer, and light sensor, respectively.

To determine the accuracy of the mock satellite attitude estimate, it was necessary to know its actual attitude. It was determined that a rotation table with a high count encoder would spin the mock satellite. The encoder would provide the actual attitude of the mock satellite.

1.4.2 Magnetometer Calibration

The magnetometer on the mock satellite had to be accurately calibrated for the mock satellite to determine an accurate attitude estimate. The magnetometer on the mock satellite is made by Dowty, it outputs three voltages and came with nominal factory settings. Without factory calibration specifications, it was found that the nominal factory settings were not accurate enough to meet the necessary requirements.

The magnetometer was calibrated by using the Helmholtz cage to generate fifty different magnetic fields. The magnetic fields were known by measuring them with a precision instrumentation magnetometer. A least squares fit was then used to determine the sesnitivity and bias of the Dowty magnetometer using the assumed known magnetic fields.

1.4.3 Attitude Determination Tests

To conduct the single degree of freedom attitude determination tests, it was first necessary to create an inertial environment for the test. The self-contained mock satellite was mounted on the rotation table in the middle of the Helmholtz cage. The laser and strobe that were used to excite the sun sensor and light sensor were then set up. The rotation table was then spun while the data from the mock satellite sensors and the encoder was collected.

1.4.4 Determine Experimental Attitude and Corresponding Error

The estimated attitude was found by post processing the data that was collected by the mock satellite during the tests. This estimated attitude was compared to the attitude found by the encoder. More accurate attitude estimates were then found by analyzing the data using differing processing methods.

1.5 Overview of Paper

Background information is given along with a literature review in Chapter 2. Then the actual hardware used in this experimental thesis is described in Chapter 3. The coordinate systems and the models used for the sensors are then presented in Chapter 4. Then the calibration process for the hardware is described in Chapter 5. The algorithms used to find the attitude are then described in Chapter 6. Then the results that were found are presented in Chapter 7. Finally, the conclusions made by the author are presented in Chapter 8.

Chapter 2

Background and Literature Review

2.1 Spacecraft Attitude

Wertz defined the attitude of a spacecraft as its orientation in space [1]. The attitude knowledge of a spacecraft can be defined by a three dimensional rotation between the spacecraft's Body Fixed Coordinate System (BFCS) and the Earth Centered Inertial (ECI) frame. The spacecraft's BFCS can be defined by three fixed orthogonal axes aligned about desired spacecraft axes.

There are many different coordinate systems in common use for the Earth, which include both inertial and Earth fixed coordinate systems. One of the most commonly used ECI coordinate systems is the J2000. The X axis of the J2000 is defined by the mean vernal equinox, the Z axis is defined by the spin axis of the earth as of January 1, 2000. The Y axis completes the right-handed triad. Vallado has gone into great detail of the definition of different coordinate systems [2].

2.2 Attitude Representation

The three dimensional rotation between two sets of coordinates can be described by rotation matrices, Euler angles, angle-axis, and quaternions [1, 3, 4].

2.2.1 Rotation Matrix

A rotation matrix is a direction cosine matrix relating the representation of a vector in one coordinate system to another. For this paper a rotation matrix will be represented with the letter "R". When dealing with three dimensional space, a rotation matrix is a 3x3 matrix. A rotation matrix is a proper real orthonormal matrix [1]. As such, its inverse is equal to its transpose, so $R^{-1} = R^T$. This property simplifies the calculation of the inverse

of a rotation matrix.

In this paper a subscript and a superscript to the left of a rotation matrix will indicate from which coordinate system and to which coordinate system the rotation matrix will rotate a vector. ${}^{I}_{b}R$ is the rotation matrix that rotates a vector described in body coordinates to inertial coordinate system. Premultipling a vector by the correct rotation matrix will represent a vector in a desired coordinate system. The orthogonality of a rotation matrix makes it easy to find a rotation matrix that rotates a vector in the opposite direction of a given rotation matrix. This is shown in Equations 2.1, 2.2, and 2.3. The Euler angles can be used to define a rotation matrix.

$$ {}^{I}r = {}^{I}_{b}R * {}^{b}r \tag{2.1} $$

$$ {}^{b}r = {}^{I}_{b}R^{T} * {}^{I}r \tag{2.2} $$

$$ {}^{b}r = {}^{b}_{I}R * {}^{I}r \tag{2.3} $$

2.2.2 Euler Angles

Euler angles, named after Leonhard Euler, are used to describe the orientation of a rigid body in three dimensional space. Three angles are needed to describe any possible orientation. For this thesis Euler angles will be represented with ϕ, θ, and ψ, they represent the rotations about the X, Y, and Z axes respectively. For this thesis the rotation sequence that will be used is ZYX. Also, for this thesis the axes that the Euler angles are rotated about are connected to the body.

2.2.3 Angle-Axis

Any possible orientation can be achieved from a given orientation by rotating the given orientation about a specific axis and through a certain angle. An "e" and α are generally

used to represent the axis and angle respectively.

2.2.4 Quaternion

Quaternions were first described by William Rowan Hamilton. Quaternions are hyper complex numbers. A "q" is used to represent a quaternion. The quaternion is defined in Equation 2.4. Equation 2.5 shows equalities that quaternions satisfy.

$$\mathbf{q} = q_4 + \mathbf{i}q_3 + \mathbf{j}q_2 + \mathbf{k}q_1 \tag{2.4}$$

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{2.5}$$

A quaternion can represent a three dimensional rotation. A quaternion is very closely related to the angle axis representation. To use a quaternion as a three dimensional rotation, its components are defined in Equations 2.6 to 2.9. An advantage in using a quaternion to represent a three dimensional rotation is that quaternions don't have problems with singularities like Euler angles and rotation matrices do. It is necessary to have at least four parameters to specify any possible attitude [5].

$$q_1 = e_1 sin(\frac{\alpha}{2}) \tag{2.6}$$

$$q_2 = e_2 sin(\frac{\alpha}{2}) \tag{2.7}$$

$$q_3 = e_3 sin(\frac{\alpha}{2}) \tag{2.8}$$

$$q_4 = cos(\frac{\alpha}{2}) \tag{2.9}$$

2.3 Attitude Determination

A minimum of two known vectors in the ECI coordinate system need to be measured by the sensors for an ADS to deterministically determine the attitude. The ADS software then compares these sensor measurements to the known vectors to determine the spacecraft's attitude as defined by a rotation matrix, or equivalently a quaternion.

2.4 Attitude Determination Methods

In general there are three possible ways that an ADS can solve for an attitude solution: deterministically, using a statistically based minimization method, or integrating the previous method with the dynamics of the spacecraft. The TRIAD method is primarily used for deterministic solutions. Statistical methods are based on solving the Wahba Problem. The integration of the dynamics is based on the Kalman Filter.

2.4.1 Deterministic Attitude Solution - Triaxis Attitude Determination (TRIAD)

The Triaxis Attitude Determination (TRIAD) algorithm was first presented by Black in 1964 [6] and is summarized by Shuster in 2006 [7]. Unlike all the other methods described in the following paragraphs, the TRIAD method requires exactly two vector measurements. It is also the simplest out of all the other methods presented, requiring the lest computation effort.

To determine the attitude the TRIAD algorithm defines three different coordinate systems and finds the rotation matrices that rotate vectors between them. The procedure to find the first two rotation matrices is shown in Equations 2.10 through 2.13. Equation 2.14 shows that the third rotation matrix is found by multiplying the first two matrices together. The TRIAD method defines the body, intermediate, and inertial coordinate systems, they are represented with b, int, and I respectively. The TRIAD algorithm is one of the methods used to solve for the attitude in this thesis. In Equations 2.10 through 2.14 $\tilde{\mathbf{b}}_1$ and $\tilde{\mathbf{b}}_2$ are vector measurements in body coordinates, $\mathbf{r}_1$ and $\mathbf{r}_2$ are the vectors that are being measured but in inertial coordinates.

$$ {}^{b}\mathbf{i} = \frac{\tilde{\mathbf{b}}_1}{|\tilde{\mathbf{b}}_1|} \qquad {}^{I}\mathbf{i} = \frac{\mathbf{r}_1}{|\mathbf{r}_1|} \tag{2.10} $$

$$ {}^{b}\mathbf{j} = \frac{\tilde{\mathbf{b}}_1 \times \tilde{\mathbf{b}}_2}{|\tilde{\mathbf{b}}_1 \times \tilde{\mathbf{b}}_2|} \qquad {}^{I}\mathbf{j} = \frac{\mathbf{r}_1 \times \mathbf{r}_2}{|\mathbf{r}_1 \times \mathbf{r}_2|} \tag{2.11} $$

$$ {}^{b}\mathbf{k} = {}^{b}\mathbf{i} \times {}^{b}\mathbf{j} \qquad {}^{I}\mathbf{k} = {}^{I}\mathbf{i} \times {}^{I}\mathbf{j} \tag{2.12} $$

$$ {}^{int.}_{b}R = [{}^{b}\mathbf{i}\ {}^{b}\mathbf{j}\ {}^{b}\mathbf{k}] \qquad {}^{int.}_{I}R = [{}^{I}\mathbf{i}\ {}^{I}\mathbf{j}\ {}^{I}\mathbf{k}] \tag{2.13} $$

$$ {}^{I}_{b}R =^{int.}_{I} R^{T} *^{int.}_{b} R \tag{2.14} $$

2.4.2 Statistical Methods - The Wahba Problem

The Wahba problem was first presented by Grace Wahba in 1965 [8]. The Wahba problem seeks to minimize the cost function shown in Equation 2.15. In Equation 2.15 $\tilde{\mathbf{b}}_j$ are the measured unit vectors in the spacecraft body frame and $\mathbf{r}_j$ are the known unit vectors in the reference frame. In most cases the reference frame is the inertial frame. There are N different $\tilde{\mathbf{b}}$'s measured and there are N different $\mathbf{r}$'s that are known. $\hat{R}$ is the rotation matrix that rotates vectors from the reference frame to the body frame. In much of the literature a rotation matrix is represented with an A, this paper will use an R instead. The variable a_j in the Equation is a weighting matrix that gives a weight to each measurement. The Wahba problem is minimized by finding the "best" rotation matrix that will rotate the vector in the reference frame into the body frame. In Equation 2.16 ν is the noise on the sensor that made the measurement.

$$ J(A) = \frac{1}{2}\sum_{j=1}^{N} a_j ||\tilde{\mathbf{b}}_j - \hat{R}\mathbf{r}_j||^2 \tag{2.15} $$

$$ \tilde{\mathbf{b}}_j = \hat{R}\mathbf{r}_j + \nu_j \tag{2.16} $$

Due to the fact that $\hat{R}$ is a rotation matrix, it is an orthonormal matrix. The orthonormality of $\hat{R}$ causes it to be subject to the following constraint: $\hat{R}^T\hat{R} = I_{3x3}$. The problem of minimizing Equation 2.15 is not trivial. There has been a lot of time and effort spent developing algorithms that solve the Wahba problem. Some of the algorithms that can be used to solve the Wahba problem are presented in the following paragraphs. All of the methods require a minimum of two vector measurements and can use as many vector measurements as are available.

Q-Method

The quaternion method or q-method was developed by Davenport and was presented in

1978 [1]. Instead of solving for a rotation matrix to minimize the Wahba cost function, the problem is modified to identify the quaternion defining the rotation. The q-method solves the Wahba problem as an eigenvalue problem. The q-method is one of the methods used to determine the estimated attitude in this thesis. The complexity and robustness of this method are addressed in several variations on the solution of this problem.

QUEST

The QUEST (QUaternion ESTimator) algorithm was developed by Shuster and Oh and was presented in 1981 [9]. The QUEST method essentially solves the q-method without calculating the eigenvalues explicitly. Because the eigenvalues are not solved for explicitly, the QUEST algorithm is more computationally efficient than the q-method, but at the same time it is less robust. Iterations are necessary when using the QUEST algorithm, Newton-Raphson iteration is commonly used. The QUEST method is one of the most popular and efficient methods to find the attitude estimate [10].

Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) method was developed by Markley and was presented in 1988 [11]. Unlike the previous two methods described, the SVD method solves directly for the rotation matrix and not the quaternion. The SVD method is more robust than the QUEST algorithm but like the q-method it is more computationally expensive than the QUEST method.

Fast Optimal Attitude Matrix (FOAM)

The Fast Optimal Attitude Matrix (FOAM) method was also developed by Markley and was presented in 1993 [12]. The FOAM method solves for the optimal attitude matrix without performing the singular value decomposition. The FOAM method is more computationally efficient than the SVD method but it is also less robust. The FOAM method also requires iterations. Markley performed a comparison of the QUEST method and the FOAM method and found that the QUEST and FOAM methods have comparable accura-

cies. Incidentally he also found that the QUEST method is less robust for the cases that he tested [12].

Estimator of Optimal Quaternion 1 (ESOQ-1) [13] and 2 (ESOQ-2) [14]

The Estimator of Optimal Quaternion 1 (ESOQ-1) method and the Estimator of Optimal Quaternion 2 (ESOQ-2) were both presented by Mortari in 1997 [13] [14]. Along with the QUEST method, ESOQ-2 is also considered one of the fastest methods to find the attitude estimate [15].

2.4.3 Filter Attitude Solution

The most common filter used in attitude determination is the Kalman filter. A Kalman smoother can also be used to find a more accurate estimate of the attitude.

Kalman Filter

The Kalman filter was described by Rudolf E. Kalman in a paper that he published in 1960 [16]. It was shown how the Kalman filter can be used in attitude determination in 1982 by Lefferts, Markley, and Shuster [17]. The Kalman filter requires much more computation than solving for the attitude solution deterministically. The Kalman filter requires a dynamic model of the system and an initial covariance matrix. The covariance matrix is a matrix of the variances and cross correlations of all the different states of the system.

The Kalman filter has the ability to filter out noise on all of the sensors. It also does not require two known independent vectors to determine the attitude like a deterministic and statistical solutions do.

Unlike the deterministic and statistical methods used to solve for the attitude, the Kalman filter takes into account not only the current measurements, but also all the previous measurements. Some spacecraft do not use the Kalman filter because of the computation expense. The Kalman filter will be able to give the most accurate real time attitude estimate. The accuracy of the attitude estimate can be more important than the additional computation requirements that accompany the Kalman filter.

Kalman Smoother

A Kalman smoother is a batch processing technique that combines the forward time Kalman filter solution with the backwards time solution, based on the relative covariances of the two estimates [18, 19]. This method typically determines a more accurate attitude estimate than the Kalman filter. This is accomplished by both forward and backward propagation through the data. The uncertainty at each measurement is reduced from the Kalman filter because not only are all the previous data samples used to better the estimate, but also the future ones. A Kalman smoother can only be used to post-process the data.

2.5 Attitude Sensors

Sensors commonly used in ADSs include: magnetometer, horizon sensor, sun sensor, differential GPS, and star sensor. The accuracies of these different sensors are shown in Table 2.1 [5]. The accuracy of all these sensors depends not only on the accuracy of the sensor itself, but also how well-known the vector is that the sensor is measuring.

2.5.1 Magnetometer

The earths magnetic field can be closely approximated by a magnetic dipole [20]. The magnetic field has a maximum magnetic intensity of 60,000 nT at the poles and a minimum magnetic intensity of 30,000 nT at the equator. The Earth's magnetic field is created by the flow of the liquid outer core of the Earth. The occurrence of the Earth's magnetic field is described by dynamo theory and was first presented by Elsasser in 1956 [21].

There are various models of the Earth's magnetic field. The most commonly used model is the International Geomagnetic Reference Field (IGRF), the IGRF model was described by Fabiano in 1969 [22]. The IGRF is a spherical harmonic model with coefficients that are updated every five years, the latest update was made in 2010 [23]. Another model of the Earth's magnetic field is the World Magnetic Model 2010 (WMM2010) [24]. This is the magnetic field model that is used by SDL. Although the models of the Earth's magnetic field are very close to the actual magnetic field, none of the models are perfect. This is because the Earth's magnetic field will compress and expand due to the intensity of solar wind.

Table 2.1: Accuracy of Attitude Sensors

Sensors	Performance	Weight (Kg)	Power (Watt)	Characteristics
Horizon Sensors	$0.05° - 1°$ (LEO)	2-5	0.3-10	Horizon uncertainties dominate accuracy. High accuracy units use scanning
Sun Sensors	$0.005° - 4°$	0.5-2	0-3	Typical field of view 60°, coarse sensor uncertainty dominated by Albedo
Star Sensors	1 arc sec - 1arc min	3-7	4-20	Typical field of view 15°, limited angular rates
Magnetometer	$0.5° - 5°$	0.6-1.2	<1	Magnetic field uncertainty is the dominating error source.
Differential GPS	$0.5° - 1°$	3	3.5-10	Main uncertainty due to multipath and error in carrier phase tracking

There are various different types of magnetometers. The type of magnetometer that is used in this thesis is a three axis fluxgate magnetometer. Chetty refers to the fluxgate magnetometer as "the transducer that converts a magnetic field into an electric voltage" [25]. There are multiple designs for a fluxgate magnetometer, but they all include a ferrous material core and an inner and outer coil of wire that are rapped around the ferrous material. The fluxgate magnetometer works by applying an alternating current to the inner coil of wire, this magnetizes the ferrous material. The magnetic field that is generated by the ferrous material will be alternate directions due to the alternating current. This constant changing of magnetic field generates an electric current in the outer coil of wire. The electric current that is produced in the outer coil is proportional to the ambient magnetic field. One set of ferrous material and wires will only be able to measure the ambient magnetic field for one axis. For a three axis fluxgate magnetometer, three different sets of ferrous material and wires are needed.

Magnetometers are a very common sensor to use in an attitude determination system. Their cost, weight, and power consumption are either the lowest or close to the lowest of all the attitude sensors. However, they do not give a very accurate vector measurement. One of the causes of the low accuracy of the magnetometer is that the magnetic field in inertial coordinates is not accurately known due to the fluctuations in the Earth's magnetic field.

The measurements made by a magnetometer can be corrupted by the electronics on the spacecraft. The use of a magnetometer in an ADS is most commonly used by spacecraft in low Earth orbits. This is because in a low Earth orbit the magnetic field is well known and has a large magnitude [5].

When magnetic attitude control is used, either as a primary control source or for momentum dumping, the sensor is susceptible to a large amount of magnetic contamination. Magnetometer measurements are not only corrupted when the rods or coils are firing, but the spacecraft can also become magnetized. This can cause the bias on the magnetometer measurements to change throughout the life of the spacecraft.

2.5.2 Sun Sensor

A sun sensor measures the vector to the center of the Sun by detecting the visible light from the Sun [1, 3, 5, 26, 27]. There are many different types of sun sensors that vary in sophistication and accuracy. There are simple sun sensors that just determine whether or not if the Sun is in the field of view. There are complex sun sensors that have a small hole or narrow slit that will only allow in a thin beam of sun light. That beam of light will only rest on some of the photo cells in the bank of photo cells behind the hole or slit. The vector to the Sun could then be calculated fairly accurately. One of the disadvantages of a sun sensor is that for most of the possible orbits of a spacecraft it will be in the umbra of the Earth some of the time.

2.5.3 Horizon Sensor

Horizon sensors, or earth sensors, measure the vector that goes from the spacecraft to the center of the Earth [1, 3, 5, 26]. The center of the Earth is found by sensing its horizon.

There is a large difference in the temperature between the atmosphere around the Earth and that of space. Usually an infrared sensor is used in a horizon sensor to detect this difference in temperature of the $\mathbf{CO_2}$ layer approximately 100 km above the Earth.

There are both static and scanning horizon sensors. Static horizon sensors generally view the entire Earth and are typically used for GEO spacecraft. The center of the Earth can then be determined by centroiding the image. Scanning horizon sensors can scan mechanically or by utilizing the spinning of a spacecraft. Scanning sensors usually have a narrow field of view. The sensor can detect the abrupt changes from when it is looking at space or at the Earth and is typically used on LEO spacecraft. Horizon sensors are commonly used on nadir pointing spacecraft so that the nadir vector can easily be detected.

2.5.4 Differential GPS

It is not only possible to use GPS for orbit determination but also for attitude determination [5, 26]. To use GPS for attitude determination it is necessary to have a minimum of two receivers. The accuracy of the vector measurement is dependent on the baseline between the GPS receivers. In order to increase the baseline the GPS receivers can be placed on booms, but the longer the booms are, the more the booms will flex which will cause the accuracy to decrease. Typical antenna differences of over 0.5 m are needed for minimal accuracy. The two GPS receivers use the phase difference between them to create a vector measurement. This method of finding a vector measurement does have a problem with integer ambiguity. There is also the problem of multi-path, this is when the GPS signal reflects off the body of the spacecraft.

2.5.5 Star Sensors

Star sensors are generally the most accurate of all the attitude sensors [1, 3, 5, 26]. The vectors to many different stars are known very accurately and the vector in body coordinates to the same star can be measured with the star sensor. Star sensors can be broken into two categories: trackers and scanners. Star trackers are used on three-axis stabilized spacecraft. Star scanners are used on spinning spacecraft. Star sensors need to have a catalog of stars

and they also need to have the ability to match stellar images to the catalog of stars. Star sensors cannot be used on spacecraft with large angular velocities. Large angular velocities causes smears in the images that the star sensor is trying to process.

2.6 Ground Testing

To ensure that a spacecraft is able to successfully complete its mission the highest degree of ground testing feasible is completed. It is necessary to check all the systems of a spacecraft prior to launch. There are different methods used to ground test the ADS for a spacecraft, including: simulation, experimental testing, and hardware in the loop simulation.

2.6.1 Simulation

A full system check of an attitude determination and control system prior to launch is very difficult to do due to the 1 g environment that is found on the surface of the Earth. The control system used while the spacecraft is in space will not be able to move the spacecraft until it is orbiting the Earth. Because of this, the full system flight check for a spacecraft's ADS will need to be at least partially completed through simulation.

Checking the flight readiness of a spacecraft is not the only reason to have a working simulation for a spacecraft. It is also beneficial to have a simulation of the spacecraft so that the it is possible to train the operational staff. It is also extremely beneficial to have a working simulation to help with problem solving if there is any type of system failure while the spacecraft is in orbit [26].

2.6.2 Experimental Testing

All the individual components on a satellite need to be experimentally checked prior to flight. Many of the components require only simple testing. The systems that don't require micro gravity to function properly are much easier to experimentally test. For example, an experimental test to check if the solar panel power generation meets design requirements is not very complicated. To complete this test it would only require putting the solar panels in direct sunlight, or under solar lamps and checking the power generation. All of the systems

that do require micro gravity to function are not as straightforward.

Mortari presented how a ground test for a moon-sun sensor was conducted in 1999 [28]. The procedure of how an ADS can be experimentally tested is explained in this thesis. One of the biggest obstacles when testing and using attitude sensors is mounting the sensor in the correct orientation. It is not uncommon to have to mount the sensor with precision optical tools, such as an auto-collimator.

2.6.3 Hardware-in-the-Loop Simulation

Hardware-in-the-Loop (HIL) simulation is a sort of combination of simulation and experimental testing. In a HIL simulation some parts of the system are simulated and some parts of the hardware are tested. There are many possible ways to set up a hardware in the loop simulator. An example of a hardware in the loop simulator for on orbit docking was presented by Chang in 2011 [29].

Part of the Attitude Determination and Control System (ADCS) can be tested using hardware in the loop simulation as well. A possible way to do that is to hook up the actuators that will be used to control the spacecraft, and then simulate the sensor outputs and the flight dynamics of the spacecraft. External sensors that will not be used in flight can then be used to check if the actuators of the spacecraft are firing correctly.

Chapter 3

Hardware and Testing

The Nano-Satellite Operation Verification and Assessment (NOVA) test facility was designed and built at SDL. The hardware that is in NOVA lab to conduct ADS testing can be separated into the hardware that emulates the inertial space, and the hardware that simulates the spacecraft's ADS (mock satellite).

3.1 Inertial Emulation Test Hardware

The hardware that emulates the inertial space consists of the inertial computer, Helmholtz cage, inertial magnetometer, rotation table, sun simulator, strobe, and inertial light sensor.

3.1.1 Inertial Computer

The inertial computer controls and records data from the Helmholtz cage and rotation table. The controllers for the Helmholtz cage and the rotation table are both in the same Simulink model in the inertial computer. The inertial computer also logs data from the inertial light sensor.

The inertial computer runs Matlab's XPC Target, which is a real-time interface, in order to control the inertial hardware and to acquire the data from the inertial hardware. The data acquisition card that the inertial computer has is made by Measurement Computing Corporation, the model number is PCIM_DAS1602/16. Four of the eight inertial DAQ card inputs are used for the tests, Table 3.1 shows how the card connects to the inertial hardware.

3.1.2 Helmholtz Cage

The Helmholtz cage is a controlled magnetic chamber, it was designed and built at SDL under the direction of Lynn Chidester. A picture of the Helmholtz cage is shown in Figure

Table 3.1: Inertial Computer DAQ Card Connections

A/D Channel	Inertial Sensor (V)
0	Inertial Magnetometer X
1	Inertial Magnetometer Y
2	Inertial Magnetometer Z
7	Inertial Light Sensor

3.1. The center of the cage is designed to have an 8 ft^3 working volume, which is a near uniform magnetic field [30]. The controller uses three axis control using feedback from the MEDA magnetometer and power switching to generate the magnetic field desired by the user.

There are three different sets of orthogonal square coils on the Helmholtz cage. Each set of coils consists of two coils. Each individual axis of the magnetic field is controlled by one of the three different sets of coils. For the Helmholtz cage to operate with the current controller it is necessary to orient the inertial magnetometer so that its sides are parallel with the sides of the cage. Due to the size of the relays that are used to control the Helmholtz cage, it is necessary to command a magnetic field that is not very close to the ambient magnetic field. If the desired magnetic field is too close to the ambient magnetic field, the relays will toggle and the magnetic field will be far from stable.

3.1.3 Inertial Magnetometer

The inertial magnetometer is made by MEDA and the model number is FVM 400. The inertial magnetometer is NIST-traceable. According to the MEDA the calibration error for the inertial magnetometer is less that 6 nT. There are two inertial magnetometers in the NOVA lab and they are only one serial number apart. A Zero Gauss chamber is used to find the bias on the inertial magnetometer. The two different MEDA magnetometers each have calibrated conditioning electronics which power the magnetometers and calibrates the measurements.

Only one inertial magnetometer is needed to run the tests, but a second was needed for the body magnetometer calibration. The axes of the Meda magnetometer are well labeled

Fig. 3.1: Helmholtz cage.

Fig. 3.2: Rotation table.

on the body of the magnetometer. Measurements from the inertial magnetometer are not actually used in the ADS. The inertial magnetic field measurements used in the ADS come from the body magnetometer while the rotation table is at zero.

3.1.4 Rotation Table

The rotation table is made by Applimotion Inc. and the model number is HO-63-B-44-A-E-000. A picture of the rotation table is shown in Figure 3.2. The rotation table has an encoder with 288,000 counts, this number of counts gives the encoder position resolution down to 0.00125°.

The mock satellite is too heavy for the rotation table to turn easily. Because of this it is important for the surface that the rotation table is on to be as level as possible. If the surface is tilted too much, the motor will get stuck at the low point and not be able to turn the mock satellite.

It is necessary to initialize the orientation of the rotation table at the beginning of

every test. The rotation table has a global home that resets the counts. When power is first applied to the rotation table, it will consider whatever position it is at home. It is then necessary to manually rotate the rotation table through the global home so that the counts can be reset.

3.1.5 Simulated Sun

The sun is simulated by a helium neon laser with a power output of 2.5 mW. The laser that is used was chosen because it's specifications are close to that of the sun. The laser is made by CVI Melles Griot and the part number is 25 LHR 691-1230.

3.1.6 Strobe

The strobe is used to synchronize time measurements for the two independent processors. It is a xenon strobe and is made by Pulsar, product number 2001w. The strobe has a dial on the back to adjust the frequency that it flashes.

3.1.7 Light Sensor

The same type of light sensor is used by both the inertial computer and the mock satellite to sense the flash from the strobe. The light sensor is made by TAOS and the part number is TSL252RSM. The output voltage of the light sensor varies with the intensity of the light shinning on it. Both the rise time and fall time of the light sensor output voltage are $7\mu s$.

3.2 Mock Satellite Hardware

The different parts of the mock satellite include a real-time CPU, a sun sensor, a magnetometer, a light sensor, a stand to mount the sensors on, and a set of batteries. Pictures of the mock satellite are shown in Figures 3.6 and 3.7.

3.2.1 Real-Time CPU: PC-104

The tasks that the PC-104 performs are power distribution and the data logging. The

Table 3.2: Mock Satellite DAQ Card Connections

A to D Input	Sensor
0	Magnetometer X
1	Magnetometer Z
2	Magnetometer Y
3	Light Sensor
4	Sun Sensor Y2
5	Sun Sensor Y1
6	Sun Sensor X2
7	Sun Sensor X1

PC-104 has a Sensoray data acquisition card, the model number is 526. The only connections that are used on the card are the analog to digital connections. There are eight analog-to-digital connections on the DAQ card. Table 3.2 shows how the body sensors connect to the DAQ card on the PC-104.

All the sensors together provide a total of nine differential analog outputs, this is one more differential analog input than the PC-104 has. For this reason the nearly constant temperature measurement of the sun sensor was not recorded. These values were replaced by a constant temperature in the data analysis.

The PC-104 uses the same real time interface as the inertial computer, but it utilizes the embedded option to allow the PC-104 to collect data while not connected to another computer. The PC-104 is the target computer. The software is loaded on the PC-104 prior to the test. After the software is loaded the PC-104 will begin collecting data from the mock satellite sensors as soon it has completed its boot sequence.

3.2.2 Body Sun Sensor

The Miniature Sun Sensor is based on a low power two degree of freedom optical light position sensitive device (PSD), it was developed at SDL and is shown in Figure 3.3. The approximate field of view of the sun sensor is $\pm 45^{\circ}$, and the approximate accuracy is 0.2° per axis. The sun sensor outputs 5 analog voltages. One of the voltages is for the temperature, it is used with the other four voltages to determine the unit vector to the sun. Based on tests performed by SDL, the accuracy decreases the closer the sun is to the limits of sun

Fig. 3.3: Miniature sun sensor.

Fig. 3.4: Body magnetometer.

sensor's field of view. The actual calibration for the sun sensor is explained in Sections 5.3 and 5.5.

3.2.3 Body Magnetometer

The body Magnetometer is a commercial navigation grade three-axis fluxgate magnetometer made by Dowty, the part number is TAM7 MK2. A picture of the magnetometer shown in Figure 3.4. It came with nominal factory settings that had a bias of 2.5 volts on each channel and a scale factor of $24,000\frac{nT}{Volt}$. It was found that the nominal settings were not accurate enough so, a new calibration was conducted and is presented in Section 5.2. The alignment accuracy is $< 0.2°$ and the range of the magnetometer is $\pm 60,000nT$.

3.2.4 Stand

The PC-104 produces a significant amount of magnetic contamination. The magnetometer is placed on top of the stand to decrease the magnetic contamination in the measurements made by the magnetometer. When mounted on the stand the magnetometer is just over a foot from the PC-104. The Solid Edge model of the stand is shown in Figure 3.5. The stand was machined so the magnetometer and the sun sensor could be precisely

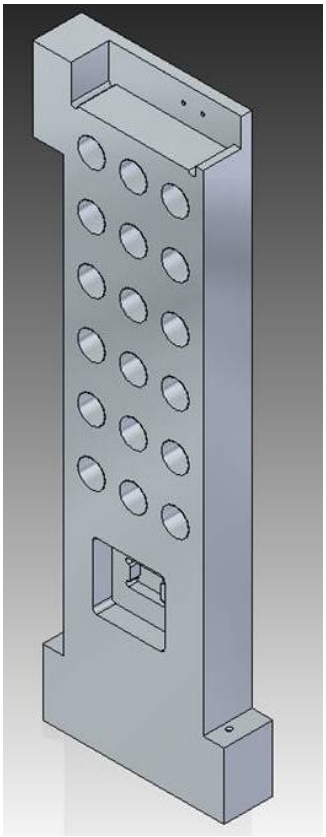

Fig. 3.5: Mock satellite stand.

mounted on it. It was designed so that the spin axis of the rotation table will go through the aperture on the sun sensor to avoid parallax effects.

3.2.5 Batteries

Batteries are needed to power the mock satellite while data is being collected. The batteries that are used on the mock satellite are salvaged batteries from spent laptop battery packs. The batteries connect to the PC-104 and the three different sensors receive power from the PC-104.

3.3 Tests

There are six different tests conducted for this experimental thesis as listed in Table 3.3. The rotation table rotated the mock satellite in two different ways. Spinning motion

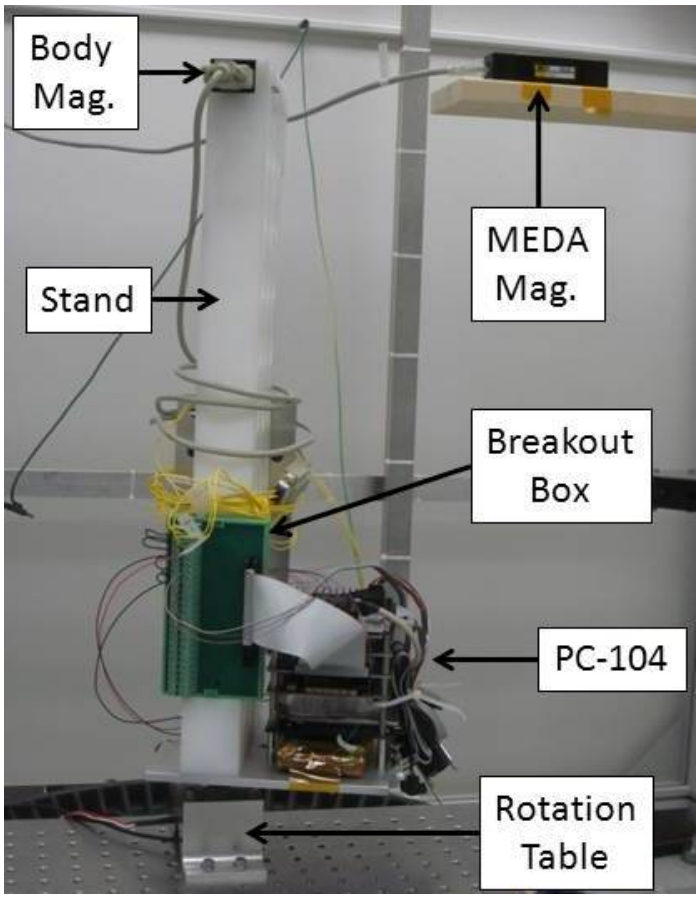

Fig. 3.6: Side view of mock satellite.

Table 3.3: The Six Different Tests

Test Name	Type of Motion	Rotation Rate	Angle Between Magnetic Field and Sun Vector
Spin 1	Spin	0.05 Hz.	Close to Orthogonal
Spin 2	Spin	0.01 Hz.	Close to Orthogonal
Spin 3	Spin	0.05 Hz.	Close to Parallel
Spin 4	Spin	0.01 Hz.	Close to Parallel
Sweep 1	Oscillate	0.06 Hz.	Close to Parallel
Sweep 2	Oscillate	0.06 Hz.	Close to Orthogonal

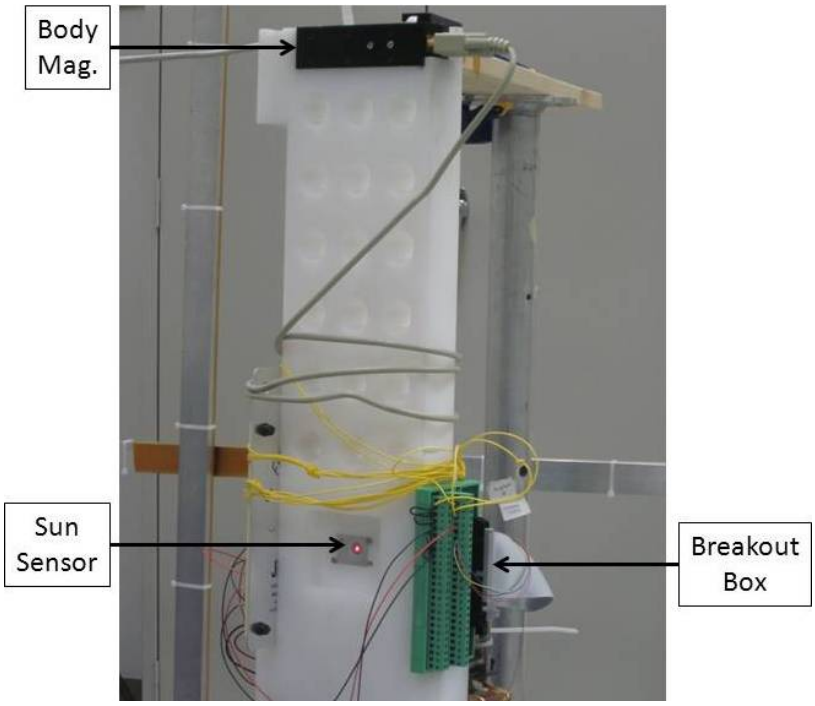

Fig. 3.7: Front view of mock satellite.

at a constant rate replicates the attitude sensor conditions found in most spin stabilized spacecraft. Non-realistic oscillatory tests were also performed to continually maintain the sun within the sensors field of view.

Chapter 4

Coordinate Systems and Sensor Models

4.1 Coordinate Systems

The coordinate systems, and the rotations between them, describe the spacecraft attitude. The coordinate systems that are used in this thesis are:

- S - The sun sensor coordinate system.
- b - The body coordinate system.
- D - The Dowty magnetometer coordinate system.
- DM - The Dowty magnetometer mounting coordinate system.
- I - The inertial coordinate system.

Figure 4.1 shows the coordinate systems used in this thesis. This is the alignment of the coordinate systems when the rotation table is at the global home. In Figure 4.1 the coordinate system DM appears to be aligned with D. This is really not the case since there is a small mounting offset between them. Figure 4.1 shows that I, B, and S are all aligned when the rotation stage is at zero, defining the nominal home orientation. In truth there is a small rotation between S and the other two. This is a result of the imperfect mounting of the sun sensor.

4.1.1 Sun Sensor Coordinates

The S coordinate system represents the mounting offset orientation of the sun sensor. The sun sensors origin is located at it's aperture. The sun sensor's X axis is aligned with the bore axis of the sun sensor. The Z axis points vertically up from the aperture. The Y axis completes the right handed coordinate system.

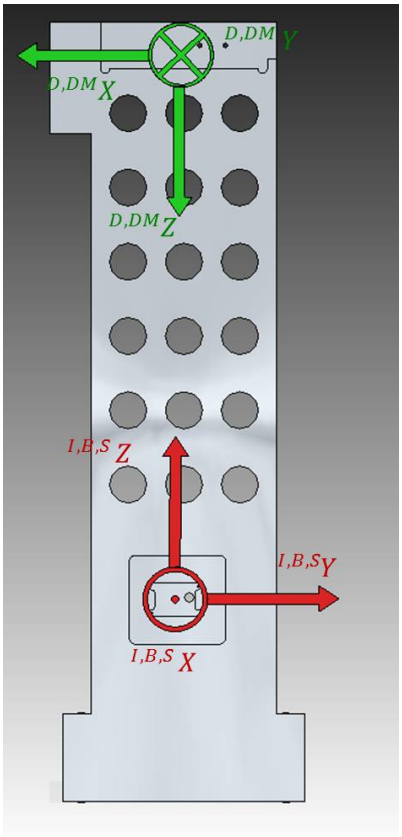

Fig. 4.1: Coordinate systems when the rotation table is at zero.

4.1.2 Body Coordinates

The origin of the body coordinates is at the aperture of the sun sensor and rotates about the inertial coordinate system. The Z axis is defined by the spin axis of the rotation table in the up direction. The X axis points away from the mock satellite and is orthogonal to the front face of the mock satellite. The Y axis completes the right handed coordinate system. The body coordinates are very close to the same as the sun sensor coordinates. The two origins are the same and all of the axes point in the same general directions. There is a small rotation due to the misalignment of the sun sensor. This misalignment is found in Section 5.5.

4.1.3 Body Magnetometer Coordinates

The D coordinate system represents the mounting offset orientation of the Dowty magnetometer. The axes of the magnetometer are defined by the manufacture with the X axis of the magnetometer pointing directly away from the side of the magnetometer that is opposite the side the cord attaches to the magnetometer. The Z axis is pointing straight down, and the Y axis completes the triad.

4.1.4 Inertial Coordinates

The inertial Z axis is also defined by the spin axis of the rotation table. The X axis is defined by the X axis of the body coordinates when the rotation table is at the global home. The Y axis completes the triad. When the rotation stage is at the global home, the inertial and body coordinates are aligned.

4.2 Attitude Determination Error

The goal of this experimental thesis is to experimentally determine the accuracy of the mock spacecraft. There is a rotation between the estimated attitude and the true attitude of the mock satellite. A quaternion or rotation matrix can represent this rotation. The error that is found is the angle of rotation between estimated attitude and the true attitude, which can be determined from the quaternion or the rotation matrix. Equation 4.1 and 4.2 show how the error quaternion and error rotation matrix are defined. In Equation 4.1 $\mathbf{q}_T$ is the true quaternion, $\hat{\mathbf{q}}$ is the estimated quaternion, and $\delta\mathbf{q}$ is the error quaternion. In Equation 4.2 ${}^I_b R_T$ is the true rotation matrix, ${}^I_b \hat{R}$ is the estimate of the rotation matrix, and $\delta^I_b R$ is the error rotation matrix.

$$\mathbf{q}_T = \hat{\mathbf{q}} * \delta\mathbf{q} \tag{4.1}$$

$${}^I_b R_T = {}^I_b \hat{R} * \delta^I_b R \tag{4.2}$$

The error quaternion and error rotation matrix can both be used to solve for the error. Equation 4.3 is found by solving for α using Equation 2.9. In Equation 4.3 $\delta\mathbf{q}_4$ is the fourth component of the error quaternion and α is the total angle of rotation from the estimated

quaternion to the true quaternion about a single axis. The error Euler angles $\delta\phi$, $\delta\theta$, $and\,\delta\psi$ come from the error rotation matrix shown in Equation 4.2.

$$\alpha = 2cos^{-1}(\delta\mathbf{q}_4) \tag{4.3}$$

$$\alpha \approx \sqrt{\delta\phi^2 + \delta\theta^2 + \delta\psi^2} \tag{4.4}$$

4.3 Attitude Truth

Equation 4.5 shows how the rotation matrix ${}^I_bR_T$ is found from the encoder. Although there is some error in the encoder, it is so much smaller than the error in the other sensors that it can be considered truth. When ${}^I_bR_T$ is calculated, by definition of the inertial coordinate system $\phi_T\,and\,\theta_T$ are both zero. ϕ_T, θ_T, $and\,\psi_T$ are the true Euler angles which come from the encoder.

$${}^I_bR_T = \begin{bmatrix} cos(\psi_T) & -sin(\psi_T) & 0 \\ sin(\psi_T) & cos(\psi_T) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

The truth model is shown in Equations 4.6 and 4.7. In the truth model ${}^I_bR_T$ is the rotation matrix that rotates a vector from body coordinates to inertial coordinates, B is the magnetic field and S is the vector to the sun. The subscript T is to indicate that it is truth.

$${}^IB = {}^I_b\,R_T * {}^bB \tag{4.6}$$

$${}^IS = {}^I_b\,R_T * {}^bS \tag{4.7}$$

4.4 Sensor Models

4.4.1 Magnetometer Sensor Model

The sensor model for the magnetometer is represented by a linear matrix model, which

allows for offset angular axes, crosstalk and bias offsets [5]. The magnetometer model is shown in Equation 4.8. On the right side of Equation 4.8 the bias is expressed as in nanoTesla instead of volts. In the magnetometer model V is the voltage from the magnetometer, V_0 is the magnetometer bias in volts, ${}^{D}B_0$ is the magnetometer bias in nanoTelsa, and V_n is the noise on the magnetometer. The bias is a negative value since it is added in the two Equations below. Q is the scale factor to convert the voltage from the magnetometer to nanoTesla, and ${}^{D}B$ is the magnetic field in Dowty coordinates. Both the scale factor and the bias for the magnetometer will have calibration errors, the models for these calibration errors are shown in Equations 4.9 and 4.10. The $\hat{}$ represents the estimate and the δ represents the error.

$$ {}^{D}B = Q * (V + V_0 + V_n) = Q * (V + V_n) +^{D} B_0 \tag{4.8} $$

$$ Q = \hat{Q} + \delta Q \tag{4.9} $$

$$ V_0 = \hat{V}_0 + \delta V_0 \tag{4.10} $$

4.4.2 Sun Sensor Model

In the sun sensor model the individual components of the vector to the sun are modeled. The sun sensor model is shown in Equation 4.11 through 4.13. In the sun sensor model S is the vector to the sun. The x, y, or z subscript to the right of S indicates the component of S as defined in the sun sensor coordinate system. The superscript S indicates that the vector is in sun sensor coordinates. The coefficients in the sun sensor model are used to relate the sun sensor voltages to the vector to the sun. X_1, X_2, Y_1, Y_2, and T form the sun sensor outputs. In Equations 4.11, and 4.12 the y and z components of the S vector divided by the x component are found. The y and z components of the S vector are found after the x component in Equation 4.13 is found. The coefficients that were provided by SDL are shown in the Appendix in Table A.1. There were also higher order models that were found

by SDL, but the linear model was found to be the most accurate.

$$\frac{{}^{S}S_y}{{}^{S}S_x} = C_{y1}X_1 + C_{y2}X_2 + C_{y3}Y_1 + C_{y4}Y_2 + C_{y5}T \tag{4.11}$$

$$\frac{{}^{S}S_z}{{}^{S}S_x} = C_{z1}X_1 + C_{z2}X_2 + C_{z3}Y_1 + C_{z4}Y_2 + C_{z5}T \tag{4.12}$$

$${}^{S}S_x = \sqrt{\frac{1}{1 + (\frac{{}^{S}S_y}{{}^{S}S_x})^2 + (\frac{{}^{S}S_z}{{}^{S}S_x})^2}} \tag{4.13}$$

Chapter 5

Calibration

5.1 Synchronizing the Body and Inertial Clocks

When comparing data collected by two different computers it is crucial that the clocks are synchronized. If the clocks are not synchronized, error due to the difference in the clocks will dominate the total error. During testing the inertial and body computers were set to sample at nominal rates of 1 kHz and 100 Hz, respectively. Through testing it was found that the actual sample rates were slightly different than these values. This could be caused by integer division of the processor clock. This difference in the two sample rates is referred to as the time scaling error. It was necessary to determine this time scaling error in the two clocks and then compensate for it.

5.1.1 Determining the Time Scaling Error

A test was conducted to determine the time scaling error between the inertial clock and body clock. In this test a high precision signal generator generated a sawtooth wave with a frequency of 1 Hz. The signal generator used was a Tektronix AFG 3252, which has an accuracy of $\pm$(1% of setting +1 mV). The sawtooth signal was then simultaneously sampled by both the inertial and body computers.

After the test, the data from both the inertial and body computers were then compared. Because the inertial and body computers did not start collecting data at the exact same time, there was a time offset between the two sets of data. The time vector on the body computer was manually aligned, so that the start of the sawtooth wave was the same on both computer time vectors. In Figure 5.1 it can be seen that the two plots are aligned at the beginning of the plot. In Figure 5.2, which is the same plot as Figure 5.1 but 268 seconds later, it can be seen that there is a large difference in the clocks of the two computers. It was

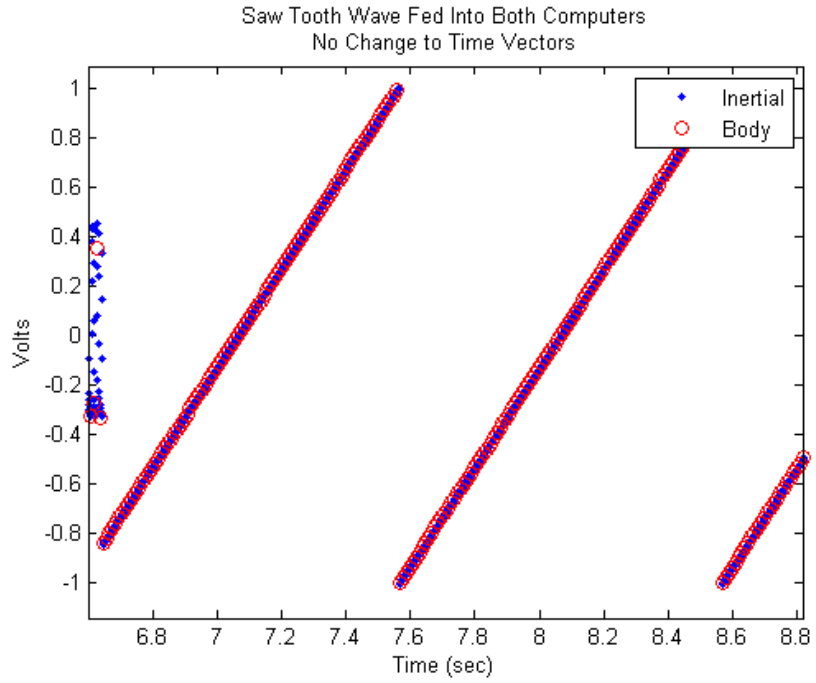

Fig. 5.1: Initial alignment of time vectors prior to time calibration.

calculated that the difference between the two clocks was about 83 ms after 268 seconds.

5.1.2 Compensate for Time Scaling Error

It was determined that it would be necessary to eliminate the time scaling error from both the inertial and body clocks. This was done by linearly scaling the time vectors for both sets of data to compensate for the time scaling error. Figure 5.2 shows that there will need to be slightly more time between samples for the body time vector and slightly less time between samples for the inertial time vector.

The amount of time necessary to add or subtract in-between samples is

$$time\ to\ adjust\ each\ sample = \frac{total\ time\ to\ adjust}{number\ of\ samples} \tag{5.1}$$

The amount of time to add in-between each sample was found for both clocks. Figure 5.3 shows that using this method did fix the time time scaling error problem.

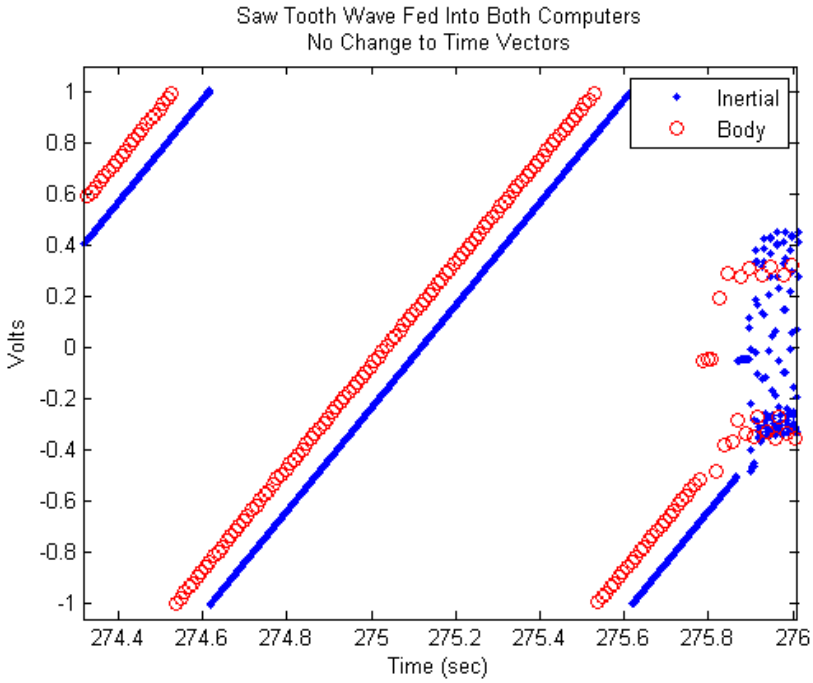

Fig. 5.2: Final alignment of time vectors prior to time calibration.

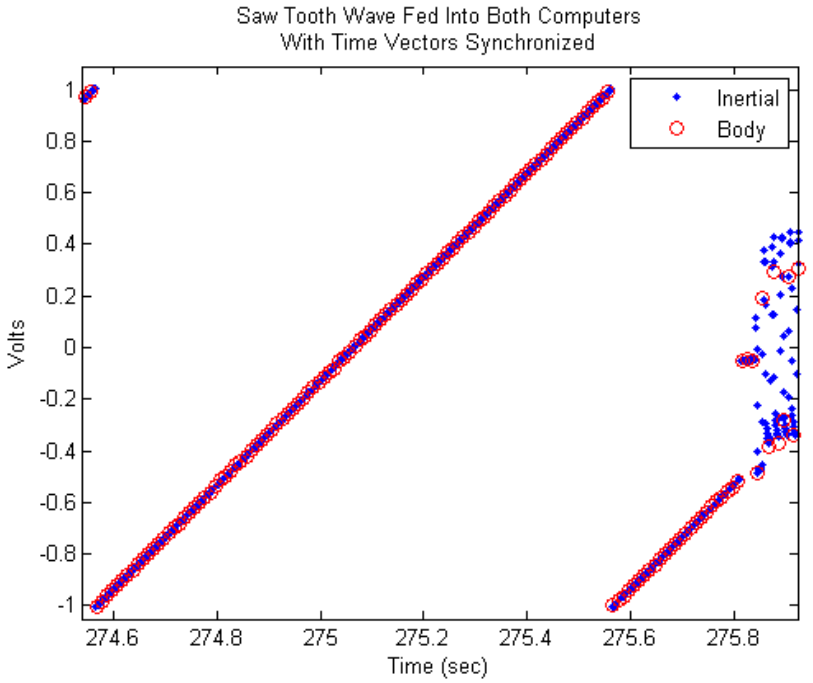

Fig. 5.3: End of plot after time scaling error compensation.

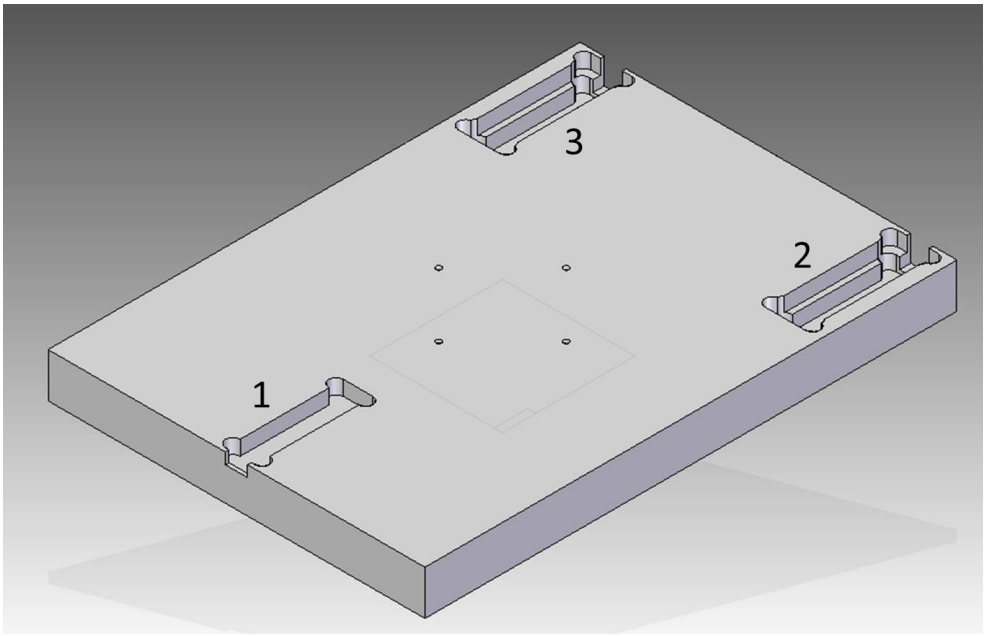

Fig. 5.4: Plastic mounting plate.

5.2 Dowty Magnetometer Calibration

A test was developed to find a new calibration for the Dowty magnetometer. Two different MEDA magnetometers and a Helmholtz cage were used to find a new calibration. One of the MEDA magnetometers was used to control the Helmholtz Cage and the other to measure the magnetic field. To distinguish between the two different MEDA magnetometers the one that controlled the Helmholtz cage will be referred to as the control MEDA and the other will be called the measure MEDA.

To maintain an accurate relative orientation between the magnetometers a plastic plate was machined, the CAD image of the plastic plate is shown in Figure 5.4. In Figure 5.4 the numbers indicate the different seats for the magnetometers. Seat one is only for the control MEDA, and seats two and three can hold both the Dowty and the measure MEDA, but not at the same time. There are nine inches between each of the seats. It was found in previous testing that the magnetometers needed to be separated by nine inches to prevent them from contaminating each others measurements.

Only seats one and two are used for the calibration test. The test developed has two different phases: a MEDA measuring phase and a Dowty measuring phase. For both of the

phases the plastic plate was placed in the middle of the Helmholtz Cage with the control MEDA in seat one.

For the first phase of the test the measure MEDA was placed in seat two. Fifty different magnetic fields were then generated using the Helmholtz cage, and each one was carefully measured by the measure MEDA. The non-uniformity of the magnetic field was as much as a 2,000 nT difference on each axis between the measure and control MEDA, which made it was necessary to use both MEDA magnetometers. This was surprising to find, and shows that the uniformity of the Helmholtz cage needs to be further explored.

For the second phase of the test, the MEDA magnetometer was taken out of seat two and the Dowty magnetometer took its place. To replicate the measurements as accurately as possible it was important that the plastic plate did not move or rotate at all while the measure MEDA was being taken out and the Dowty was being inserted. The same fifty different magnetic fields were then generated in the same order and each one was measured by the Dowty magnetometer. If the alignment was maintained the magnetic field was repeatable to ± 5 nT. After this phase of the test was completed there were voltage measurements made by the Dowty magnetometer for 50 different magnetic fields.

With the 50 voltage measurements from the Dowty and the 50 nanoTesla measurements from the measure MEDA, a calibration can be completed. A least squares fit was then used to find the biases and the sensitivities for the Dowty magnetometer. Cross coupling terms were also taken into account which allowed a sensitivity matrix to be found. Equation5.2 is the same as Equation 4.8 but with the individual components of the matrix and vectors shown. All of the measurements made for the calibration are time averages. The voltage Vn is assumed to be zero-mean white-noise, the time average of zero-mean white-noise is zero, which allows the noise to be ignored. The new organization of Equation 5.3 allows for the least squares solution of the magnetometer calibration to be found.

$$\begin{bmatrix} {}^{D}B_x \\ {}^{D}B_y \\ {}^{D}B_z \end{bmatrix} = \begin{bmatrix} Q_{xx} & Q_{xy} & Q_{xz} \\ Q_{yx} & Q_{yy} & Q_{yz} \\ Q_{zx} & Q_{zy} & Q_{zz} \end{bmatrix} * \left(\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \begin{bmatrix} V_{nx} \\ V_{ny} \\ V_{nz} \end{bmatrix} \right) + \begin{bmatrix} {}^{D}B_{x0} \\ {}^{D}B_{y0} \\ {}^{D}B_{z0} \end{bmatrix} \tag{5.2}$$

$$\begin{bmatrix} {}^{D}B_x \\ {}^{D}B_y \\ {}^{D}B_z \end{bmatrix} = \begin{bmatrix} V_x & V_y & V_z & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & V_x & V_y & V_z & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & V_x & V_y & V_z & 1 \end{bmatrix} \begin{bmatrix} Q_{xx} \\ Q_{xy} \\ Q_{xz} \\ {}^{D}B_{x0} \\ Q_{yx} \\ Q_{yy} \\ Q_{yz} \\ {}^{D}B_{y0} \\ Q_{zx} \\ Q_{zy} \\ Q_{zz} \\ {}^{D}B_{z0} \end{bmatrix} \tag{5.3}$$

If Equation 5.3 is expressed in matrix form as $B = V * \mathbf{Q}$, the least squares solution for the scale factors and the biases is:

$$\mathbf{Q} = (V^T V)^{-1} V^T B \tag{5.4}$$

$(V^T V)^{-1} V^T$ is the pseudo inverse. The units for the scale factors are $\frac{nT}{Volts}$, and the units for the biases are nT. The scale factor matrix and the bias vector estimates along with their uncertainties are shown below. The uncertainties came from the estimated covariance matrix at the one sigma level.

$$Sensitivity = \begin{bmatrix} 23,963 & 17 & -137 \\ -65 & 24,061 & -50 \\ 10 & 57 & 23,853 \end{bmatrix} (\frac{nT}{Volt})$$

$$Uncertainty\ Sensitivity = \begin{bmatrix} 18.4 & -2.7 & 1.4 \\ -2.7 & 17.8 & 1 \\ 1.4 & 1 & 27.5 \end{bmatrix} (\frac{nT}{Volt})$$

$$Bias = \begin{bmatrix} -59,198 \\ -59,949 \\ -59,997 \end{bmatrix} (nT)$$

$$Uncertainty\ Bias = \begin{bmatrix} 494.3 \\ 494.3 \\ 494.3 \end{bmatrix} (nT)$$

To verify that this calibration method worked for the Dowty magnetometer, half of the fifty different magnetic field measurements were used to create the calibration using a least squares fit. The other half of the data was used to validate the calibration. The calibration coefficients were used with the measured voltages and the resulting magnetic field estimate was compared to the field measured by the measure MEDA magnetometer.

The calibration that was found using only half of the data was just slightly different than the one that was found using all of the data. The magnetic fields that were found using the calibration on half of the data were very close to the true values. The square root of the mean of the squared errors (rms) for the values was 46.2 nT. Figure 5.5 shows the error for the calibrated data. The rms error found using the nominal factory settings was 307 nT, so the calibration found made a large improvement.

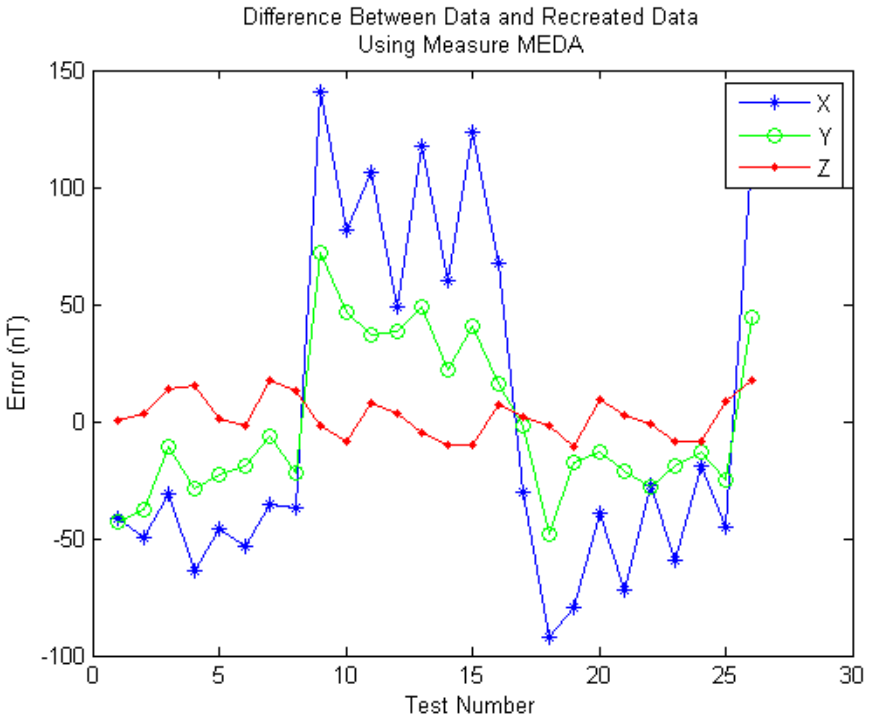

Fig. 5.5: Error in magnetic field found using new calibration.

5.3 Sun Sensor Calibration

There was a set of calibration data provided by Mitch Whiteley of SDL, which included different order polynomial fits. The first order fit was used due to its computation ease and also its accuracy, which was found in preliminary testing. One possible cause of why the higher order calibrations did not yield accurate results could have been caused by the different intensities of the laser. The same laser was used when SDL calibrated the sun sensor and when the tests for this experimental thesis were conducted, but the distance the sun sensor was from the laser was not the same. The distance the sun sensor was from the laser was much larger during the tests for this experimental thesis, which caused the intensity to decrease. The equations used to calibrate the sun sensor are defined in Equations 4.11 through 4.13.

5.4 Magnetometer Mounting Alignment

It is necessary to find the angular offset mounting of the body magnetometer. The

magnetometer was spun and the measurements were then compared with the encoder data. After the data was collected the voltage outputs were converted to nT. After the magnetic field vector was measured then each set of vector components was normalized, which created a unit vector for each measurement. Then the constant magnetometer mounting angles ϕ and θ were found. After ϕ and θ were found it was then possible to find ψ for each set of measurements, which is the rotation of the the mock satellite. This is done by only using the magnetometer, this rotation angle can then be compared to the rotation angle found by the rotation table.

5.4.1 Normalizing Measurements

Normalizing the measurements was accomplished by dividing each individual vector component by the magnitude of the vector. In Equation 5.5 B is the magnetic field measured by the body magnetometer.

$$B_{i_{normalized}} = \frac{B_i}{|B_i|} \tag{5.5}$$

This normalization process was completed for each vector measurement from the magnetometer.

5.4.2 Phi and Theta

The rotation matrix for the Dowty magnetometer is shown in Equation 5.6. The rotation matrix used is a ZYX rotation [3].

${}^I_bR$ can be used to rotate a vector from body to inertial coordinates. In this thesis c() and s() are used to represent cos() and sin().

$$^IB = \begin{bmatrix} {}^IB_x \\ {}^IB_y \\ {}^IB_z \end{bmatrix} = {}^I_b R(\psi) *^b B = \quad ... \tag{5.6}$$

$$\begin{bmatrix} c(\theta)*c(\psi) & s(\theta)*s(\phi)*c(\psi)-c(\phi)*s(\psi) & s(\theta)*c(\phi)*c(\psi)+s(\phi)*s(\psi) \\ c(\theta)*s(\psi) & s(\theta)*s(\phi)*s(\psi)+c(\phi)*c(\psi) & s(\theta)*c(\phi)*s(\psi)-s(\phi)*c(\psi) \\ -s(\theta) & c(\theta)*s(\phi) & c(\theta)*c(\phi) \end{bmatrix} * \begin{bmatrix} {}^{b}B_x \\ {}^{b}B_y \\ {}^{b}B_z \end{bmatrix}$$

The bottom row of the rotation matrix does not contain ψ. The equation for ${}^{I}B_z$ can be pulled out from the equation above.

$${}^{I}B_z = \begin{bmatrix} -s(\theta) & c(\theta)*s(\phi) & c(\theta)*c(\phi) \end{bmatrix} * \begin{bmatrix} {}^{b}B_x \\ {}^{b}B_y \\ {}^{b}B_z \end{bmatrix} \tag{5.7}$$

The equation can be rewritten.

$${}^{I}B_z = \begin{bmatrix} {}^{b}B_x & {}^{b}B_y & {}^{b}B_z \end{bmatrix} * \begin{bmatrix} -s(\theta) \\ c(\theta)*s(\phi) \\ c(\theta)*c(\phi) \end{bmatrix} \tag{5.8}$$

The above equation can be rewritten in matrix form.

$${}^{I}B_z = {}^{b}B * X \tag{5.9}$$

This equation will hold true for an entire test since ϕ and θ are constants. X can be solved for using a least squares fit. After X is solved for ϕ and θ both can be found using trigonometric functions.

$$X = \begin{bmatrix} -s(\theta) \\ c(\theta)*s(\phi) \\ c(\theta)*c(\phi) \end{bmatrix} \tag{5.10}$$

$$\theta = sin^{-1}(-X_1) \tag{5.11}$$

$$\phi = sin^{-1}(\frac{X_2}{cos(\theta)}) \tag{5.12}$$

A check for this process was done to verify that $cos(\theta)*cos(\phi) = X_3$.

Using this method, it was found that the mounting angles for the body magnetometer were $\phi = -0.15°$ and $\theta = -0.10°$, the standard deviation fro both ϕ and θ is $0.001°$. The standard deviation was found by squaring the covariance from the estimated covariance matrix.

5.4.3 Psi

The parts of the equation ${}^{I}B = {}^{I}_{b} R * {}^{b} B$ that were not used to find ϕ and θ were used to find ψ. The individual components of the matrices are defined in the Equation 5.6. The ψ found in this Section is the rotation angle for the magnetometer for each measurement made. The mounting angle ψ_0 can be found by averaging the rotation angle ψ while the rotation table is stationary at the beginning of the test.

The equations for ${}^{I}B_x$ and ${}^{I}B_y$ can be pulled off the above equation.

$$ {}^{I}B_x = ... \tag{5.13} $$

$$ \begin{bmatrix} c(\theta) * c(\psi) & s(\theta) * s(\phi) * c(\psi) - c(\phi) * s(\psi) & s(\theta) * c(\phi) * c(\psi) + s(\phi) * s(\psi) \end{bmatrix} * \begin{bmatrix} {}^{b}B_x \\ {}^{b}B_y \\ {}^{b}B_z \end{bmatrix} $$

$$ {}^{I}B_y = ... \tag{5.14} $$

$$ \begin{bmatrix} c(\theta) * s(\psi) & s(\theta) * s(\phi) * s(\psi) + c(\phi) * c(\psi) & s(\theta) * c(\phi) * s(\psi) - s(\phi) * c(\psi) \end{bmatrix} * \begin{bmatrix} {}^{b}B_x \\ {}^{b}B_y \\ {}^{b}B_z \end{bmatrix} $$

In the two equations just presented there is only one unknown. Both equations will need to be solved though because each equation will give two possible solutions. To solve the equations the matrix multiplication on the right side of the equations is first performed. Every term on the right side of the equality in the two equations is multiplied by a $\cos(\psi)$

or a $\sin(\psi)$. Isolating $\cos(\psi)$ and $\sin(\psi)$ makes it possible to solve for ψ.

$$\begin{aligned} {}^{I}B_x &= (c(\theta) * {}^{b}B_x + s(\theta) * s(\phi) * {}^{b}B_y + s(\theta) * c(\phi) * {}^{b}B_z) * c(\psi) \\ &\quad +(-c(\phi) * {}^{b}B_y + s(\phi) * {}^{b}B_z) * s(\psi) \end{aligned} \tag{5.15}$$

$$\begin{aligned} {}^{I}B_y &= (c(\phi) * {}^{b}B_y - s(\phi) * {}^{b}B_z) * c(\psi) \\ &\quad +(c(\theta) * {}^{b}B_x + s(\theta) * s(\phi) * {}^{b}B_y + s(\theta) * c(\phi) * {}^{b}B_z) * s(\psi) \end{aligned} \tag{5.16}$$

The equations can be simplified by setting a few equalities. The equalities that will simplify the equations are:

$$k = c(\theta) * {}^{b}B_x + s(\theta) * s(\phi) * {}^{b}B_y + s(\theta) * c(\phi) * {}^{b}B_z \tag{5.17}$$

$$l = -c(\phi) * {}^{b}B_y + s(\phi) * {}^{b}B_z \tag{5.18}$$

$$m = c(\phi) * {}^{b}B_y - s(\phi) * {}^{b}B_z \tag{5.19}$$

$$n = c(\theta) * {}^{b}B_x + s(\theta) * s(\phi) * {}^{b}B_y + s(\theta) * c(\phi) * {}^{b}B_z \tag{5.20}$$

Using these four equalities Equations 5.15 and 5.16 can be simplified to the following two equations.

$${}^{I}B_x = k * c(\psi) + l * s(\psi) \tag{5.21}$$

$${}^{I}B_y = m * c(\psi) + n * s(\psi) \tag{5.22}$$

With the equations in this form ψ can be solved for. Equations 5.23 and 5.24 solve for ψ [31].

$$\psi = 2tan^{-1}\left(\frac{l \pm \sqrt{l^2 + k^2 - {}^{I}B_x^2}}{k + {}^{I}B_x}\right) \tag{5.23}$$

$$\psi = 2tan^{-1}\left(\frac{n \pm \sqrt{n^2 + m^2 - {}^{I}B_y^2}}{m + {}^{I}B_y}\right) \tag{5.24}$$

The $\pm$ operator shows that both of the equations just presented will yield two different solutions. The true ψ is the solution that both of the equations find. The ψ found can then be compared to the rotation angle from the encoder to determine the accuracy of the magnetometer by itself. Using this method it was found the $\psi_0 = 0.07°$ and has a standard deviation of 0.08°.

Results

The value ψ_0 is the initial offset angle of rotation about the Z axis. Figure 5.6 shows the raw magnetometer data. The calibration that was found for the magnetometer is then used to convert the voltage output into nanoTesla. Figure 5.7 shows the calibrated output for the Dowty magnetometer. This is the magnetic field vector in the Dowty coordinates.

After the magnetometer data was calibrated the attitude was then found with the magnetometer by itself. The ψ found by the magnetometer is plotted with the ψ_T found by the encoder in Figures 5.8 and 5.9. The results from the magnetometer have an rms error of 0.55°, which is only on the one axis. Figure 5.10 is a plot of the difference between the encoder ψ_T and the magnetometer ψ.

5.5 Inertial Sun Vector and Sun Sensor Mounting Alignment

5.5.1 First Attempt for Inertial Sun Vector and Sun Sensor Mounting Alignment Calibration

A nonlinear least squares fit was used to find the inertial sun vector and the mounting alignment for the sun sensor. To find the least squares fit, first a measurement model had

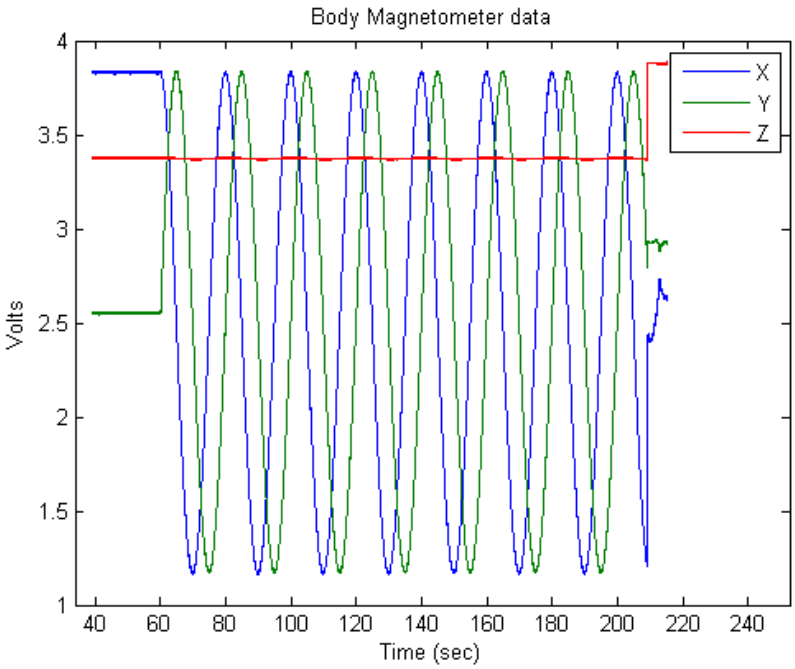

Fig. 5.6: Raw Dowty magnetometer data for Spin 1 test.

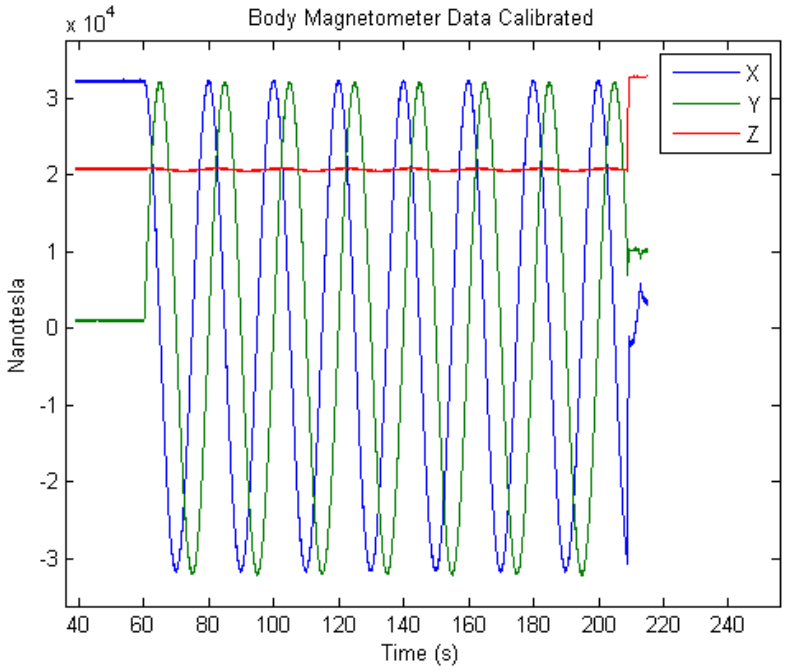

Fig. 5.7: Magnetic field vector in Dowty coordinates for Spin 1 test.

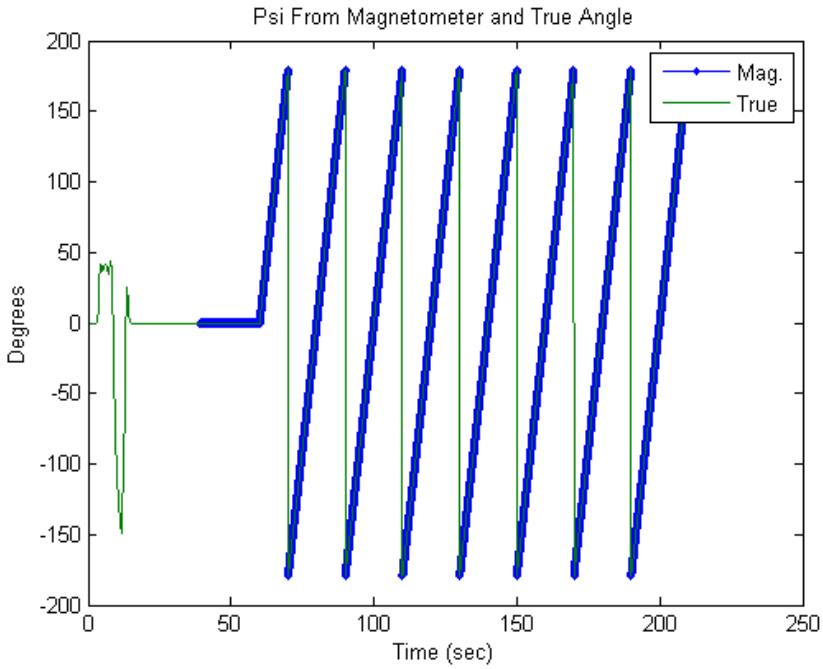

Fig. 5.8: Psi from magnetometer and encoder angle for entire Spin 1 test.

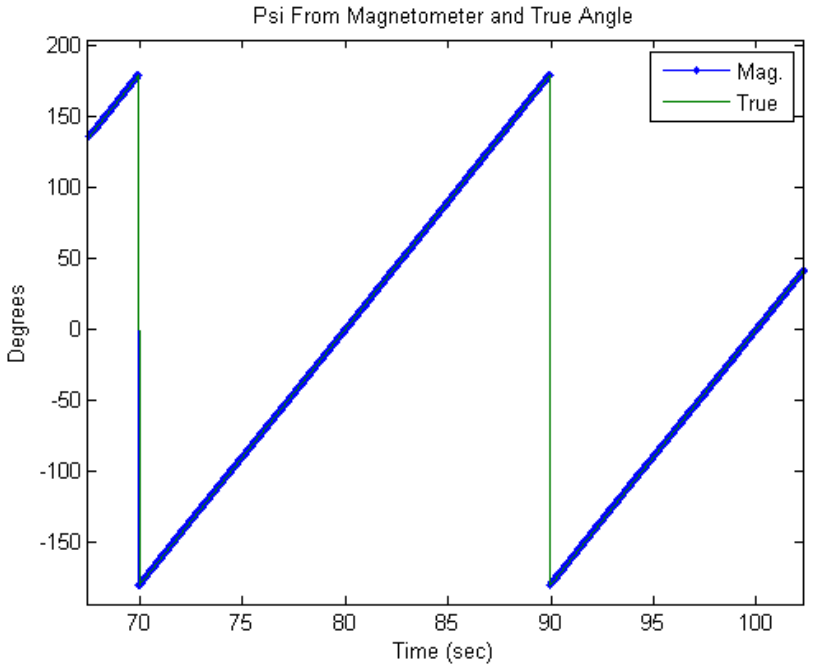

Fig. 5.9: Psi from magnetometer and encoder angle for one revolution.

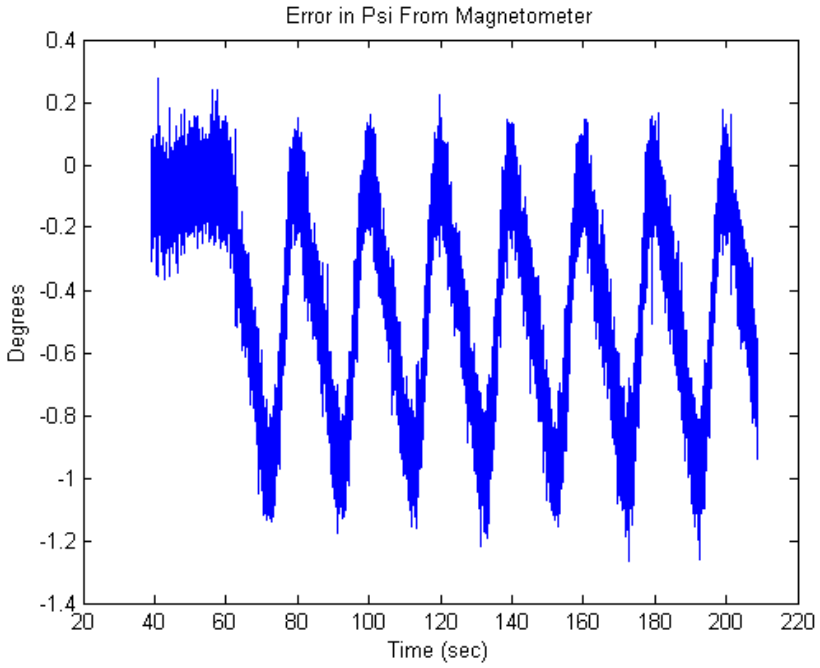

Fig. 5.10: RMS error of magnetometer ψ.

to be developed. The measurement model is shown in Equation 5.26. It is a nonlinear least squares fit because the Euler angles and the inertial sun vector are nonlinearly coupled, this is shown in the measurement model. The variable ${}^{S}S$ is sun vector in sun sensor coordinates. The superscript S is used to represent sun sensor coordinates. $\alpha = \phi, \theta, \psi$, which represents the Euler angles that rotate from body coordinates to sun sensor coordinates.

$$ {}^{B}S = {}^{b}_{S}R * {}^{S}S = {}^{b}_{I}R * {}^{I}S \qquad {}^{S}S = {}^{b}_{S}R^{T} * {}^{b}_{I}R * {}^{I}S \tag{5.25} $$

$$ {}^{S}S(\alpha, {}^{I}S) = {}^{S}_{b}R(\alpha) * {}^{b}_{I}R(\psi_T) * {}^{I}S \tag{5.26} $$

How this equation is linearized is shown in the following equation.

$$ \begin{aligned} {}^{S}S(\alpha, {}^{I}S) &= {}^{S}S(\alpha_0, {}^{I}S_0) + \frac{\partial({}^{S}_{b}R(\alpha) * {}^{b}_{I}R(\psi_T) * {}^{I}S)}{\partial \alpha}|_{\alpha_0, {}^{I}S_0}(\alpha - \alpha_0) \\ &\quad + \frac{\partial({}^{S}_{b}R(\alpha) * {}^{b}_{I}R(\psi_T) * {}^{I}S)}{\partial {}^{I}S}|_{\alpha_0, {}^{I}S_0}({}^{I}S - {}^{I}S_0) \end{aligned} \tag{5.27} $$

To calculate the partial derivatives the matrix multiplication was first completed. This way the partial derivative of a matrix with respect to a vector never had to be found. It is assumed that the rotation from sun sensor coordinates to body coordinates will be small, because of this the small angle approximation is used for the rotation matrix. The equality

$$ {}^{S}S_e = {}^{S}_{b}R(\alpha) *^{b}_{I} R(\psi_T) *^{I} S \tag{5.28} $$

will be used from now on in this thesis. The variable ${}^{S}S_e$ is the explicitly found vector to the sun in the sun sensor coordinates.

$$ {}^{S}S_e = \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} \begin{bmatrix} c(\psi_T) & -s(\psi_T) & 0 \\ s(\psi_T) & c(\psi_T) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{1 - {}^{I}S_y^2 - {}^{I}S_z^2} \\ {}^{I}S_y \\ {}^{I}S_z \end{bmatrix} \tag{5.29} $$

The vector to the sun has to be constrained to be a unit vector, which is why ${}^{I}S_x = \sqrt{1 - {}^{I}S_y^2 - {}^{I}S_z^2}$.

$$ {}^{S}S_e = \begin{bmatrix} \sqrt{1 - {}^{I}S_y^2 - {}^{I}S_z^2} * (c(\psi_T) + \psi s(\psi_T)) + {}^{I}S_y * (-s(\psi_T) + \psi c(\psi_T)) - {}^{I}S_z * \theta \\ \sqrt{1 - {}^{I}S_y^2 - {}^{I}S_z^2} * (-\psi c(\psi_T) + s(\psi_T)) + {}^{I}S_y * (\psi s(\psi_T) + c(\psi_T)) + {}^{I}S_z * \phi \\ \sqrt{1 - {}^{I}S_y^2 - {}^{I}S_z^2} * (\theta c(\psi_T) - \phi s(\psi_T)) + {}^{I}S_y * (-\theta s(\psi_T) - \phi c(\psi_T)) + {}^{I}S_z \end{bmatrix} \tag{5.30} $$

Now that ${}^{S}S_e$ is expressed as a vector the partial derivatives can be computed.

$$ \frac{\partial({}^{S}S_e)}{\partial \alpha}|_{\alpha_0, {}^{I}S_0} = \begin{bmatrix} 0 & \cdots \\ {}^{I}S_{z_0} & \cdots \\ \sqrt{1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2} * -s(\psi_T) - {}^{I}S_{y_0} * c(\psi_T) & \cdots \\ -{}^{I}S_{z_0} & \cdots \\ 0 & \cdots \\ \sqrt{1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2} * c(\psi_T) - {}^{I}S_{y_0} * s(\psi_T) & \cdots \end{bmatrix} \tag{5.31} $$

$$\begin{bmatrix} \sqrt{1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2} * s(\psi_T) + {}^{I}S_{y_0} * c(\psi_T) \\ \sqrt{1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2} * -c(\psi_T) + {}^{I}S_{y_0} * s(\psi_T) \\ 0 \end{bmatrix}$$

$$\frac{\partial({}^{S}S_e)}{\partial {}^{I}S}|_{\alpha_0,{}^{I}S_0} = \begin{bmatrix} -(1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2)^{-\frac{1}{2}} * {}^{I}S_{y_0} * (c(\psi_T) + \psi s(\psi_T)) + (-s(\psi_T) + \psi c(\psi_T)) & \cdots \\ -(1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2)^{-\frac{1}{2}} * {}^{I}S_{y_0} * (-\psi c(\psi_T) + s(\psi_T)) + (\psi s(\psi_T) + c(\psi_T)) & \cdots \\ -(1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2)^{-\frac{1}{2}} * {}^{I}S_{y_0} * (\theta c(\psi_T) - \phi s(\psi_T)) + (-\theta s(\psi_T) - \phi c(\psi_T)) & \cdots \end{bmatrix}$$

$$\begin{bmatrix} -(1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2)^{-\frac{1}{2}} * {}^{I}S_{z_0} * (c(\psi_T) + \psi s(\psi_T)) - \theta \\ -(1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2)^{-\frac{1}{2}} * {}^{I}S_{z_0} * (-\psi c(\psi_T) + s(\psi_T)) + \phi \\ -(1 - {}^{I}S_{y_0}^2 - {}^{I}S_{z_0}^2)^{-\frac{1}{2}} * {}^{I}S_{z_0} * (\theta c(\psi_T) - \phi s(\psi_T)) + 1 \end{bmatrix} \quad (5.32)$$

Now that the partial derivatives are known, the equations can be reorganized to solve for the α and ${}^{I}S$.

$${}^{S}S(\alpha, {}^{I}S) - {}^{S}S(\alpha_0, {}^{I}S_0) + \frac{\partial({}^{S}S_e)}{\partial \alpha}|_{\alpha_0,{}^{I}S_0} * \alpha_0 + \frac{\partial({}^{S}S_e)}{\partial {}^{I}S}|_{\alpha_0,{}^{I}S_0} * {}^{I}S_0 \quad = \quad ... \quad (5.33)$$

$$\frac{\partial({}^{S}S_e)}{\partial \alpha}|_{\alpha_0,{}^{I}S_0}\alpha + \frac{\partial({}^{S}S_e)}{\partial {}^{I}S}|_{\alpha_0,{}^{I}S_0}{}^{I}S$$

The left side of the above equation is known. It will be expressed as L, for left, in future equations. The only two variables in the right side of the equation that are unknown are α and ${}^{I}S$. The equation can be reorganized so that α and ${}^{I}S$ can be solved for.

$$L = \begin{bmatrix} \frac{\partial({}^{S}S_e)}{\partial \alpha}|_{\alpha_0,{}^{I}S_0} & \frac{\partial({}^{S}S_e)}{\partial {}^{I}S}|_{\alpha_0,{}^{I}S_0} \end{bmatrix} \begin{bmatrix} \alpha \\ {}^{I}S \end{bmatrix} \quad (5.34)$$

Which is explicitly expressed in Equation 5.35.

Table 5.1: Results for Inertial Sun Vector and Sun Sensor Mounting Alignment Calibration

Component	Sweep 2
${}^{I}S_x$	0.9847
${}^{I}S_y$	0.156
${}^{I}S_z$	-0.038
ϕ	0.95°
θ	2.95°
ψ	−5.36°

$$L = \begin{bmatrix} \frac{\partial({}^{S}S_e)}{\partial\alpha}|_{\alpha_0,{}^{I}S_0} & \frac{\partial({}^{S}S_e)}{\partial{}^{I}S}|_{\alpha_0,{}^{I}S_0} \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ {}^{I}S_y \\ {}^{I}S_z \end{bmatrix} \tag{5.35}$$

To solve for this nonlinear least squares problem, it is necessary to provide initial values for the Euler angles and the inertial sun vector. Iterations have to be done in order to find the best solution. To iterate, the initial approximations are replaced with the solutions to the previous iteration. After there is essentially no change in the solutions from iteration to iteration, it can be assumed that the solution has been found. After the solution has been found the sun vector in body coordinates is known.

Results

The results for this calibration are shown in Table 5.1. It was found that this calibration was not satisfactory. The Y component of the recreated sun vector in inertial coordinates is shown in Figure 5.11. If the sun sensor calibration were to work properly the sun vector in inertial coordinates would be constant. As shown in Figure 5.11 the Y component of the sun vector in inertial coordinates is not constant.

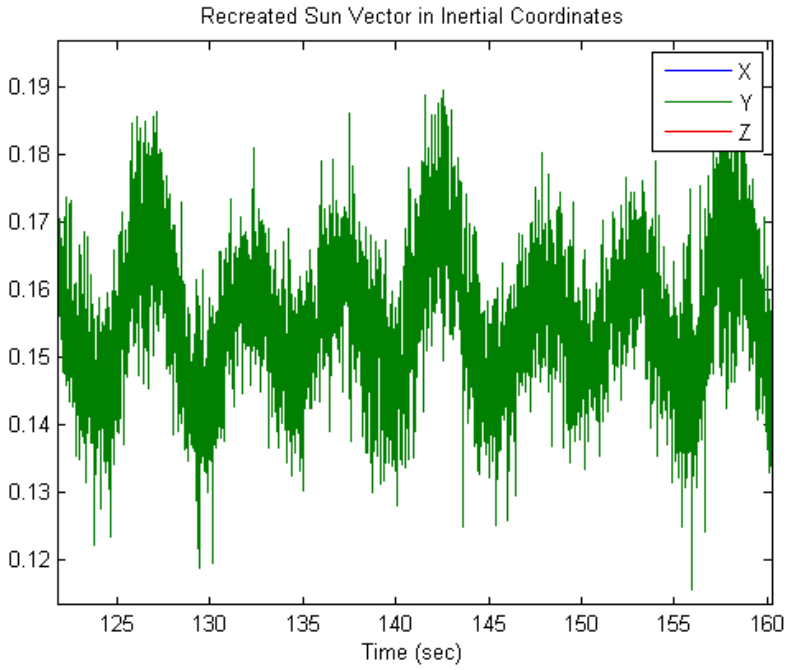

Fig. 5.11: Inertial sun vector Y component using initial linear calibration.

5.5.2 Second Attempt for Inertial Sun Vector and Sun Sensor Mounting Alignment Calibration

To improve the calibration, third order terms were added. The new calibration is shown in Equations 5.36 to 5.38. The new calibration adds third and second order calibration terms to the X and and Y components of the vector to the sun. Higher order terms were not added to the Z components due to the fact that the mock satellite is rotating about the Z axis, which causes the Z component to be fairly constant. After this new calibration was found, the same process shown in Section 5.5.1 was then used to find the inertial sun vector and the the sun sensor mounting angles.

$$\begin{aligned}\frac{{}^{S}S_y}{{}^{S}S_x} &= C_{y1}X_1 + C_{y2}X_2 + C_{y3}Y_1 + C_{y4}Y_2 + C_{y5}Temp + C_{y6}(\frac{X_1 - X_2 - Y_1 + Y_2}{X_1 + X_2 + Y_1 + Y_2})^3 \\ &+C_{y7}(\frac{X_1 - X_2 - Y_1 + Y_2}{X_1 + X_2 + Y_1 + Y_2})^2 + C_{y8}(\frac{X_1 - X_2 - Y_1 + Y_2}{X_1 + X_2 + Y_1 + Y_2}) + C_{y9} \qquad (5.36)\end{aligned}$$

Table 5.2: Results for Inertial Sun Vector and Sun Sensor Mounting Alignment Calibration

Component	Sweep 1	Sweep 1 Covariance	Sweep 2	Sweep 2 Covariance
${}^{I}S_x$	0.984	$1.7e^{-7}$	0.984	$5.5e^{-6}$
${}^{I}S_y$	0.175	$6.7e^{-8}$	0.176	$5.5e^{-6}$
${}^{I}S_z$	-0.029	$1.6e^{-7}$	-0.034	$7.5e^{-8}$
ϕ	0.86°	$7.4°e^{-7}$	0.86°	$9.9°e^{-7}$
θ	2.33°	$1.1°e^{-5}$	2.69°	$5.1°e^{-6}$
ψ	$-4.31°$	$3.9°e^{-6}$	$-4.31°$	$3.4°e^{-4}$

$$\frac{{}^{S}S_z}{{}^{S}S_x} = C_{z1}X_1 + C_{z2}X_2 + C_{z3}Y_1 + C_{z4}Y_2 + C_{z5}Temp \tag{5.37}$$

$${}^{S}S_x = \sqrt{\frac{1}{1 + (\frac{{}^{S}S_y}{{}^{S}S_x})^2 + (\frac{{}^{S}S_z}{{}^{S}S_x})^2}} \tag{5.38}$$

Results

The results of this new calibration and the estimated covariance values are shown in Table 5.2. Figure 5.12 is a plot of the Y component of the sun vector in inertial coordinates using this new calibration. As can be seen in Figure 5.12 the new calibration is an improvement. The Y component is much more constant in Figure 5.12 compared to Figure 5.11.

5.5.3 Mechanical Measurement of Inertial Sun Vector

The inertial sun vector was mechanically measured with a tape measure. To better understand the accuracy of this measurement an uncertainty analysis was conducted on this measurement. All three components of the vector to the sun in inertial coordinates were measured. The measured components of the inertial sun vector are given in Equation 5.39. The subscripts xm, ym ,and zm stand for x measurement, y measurement, and z

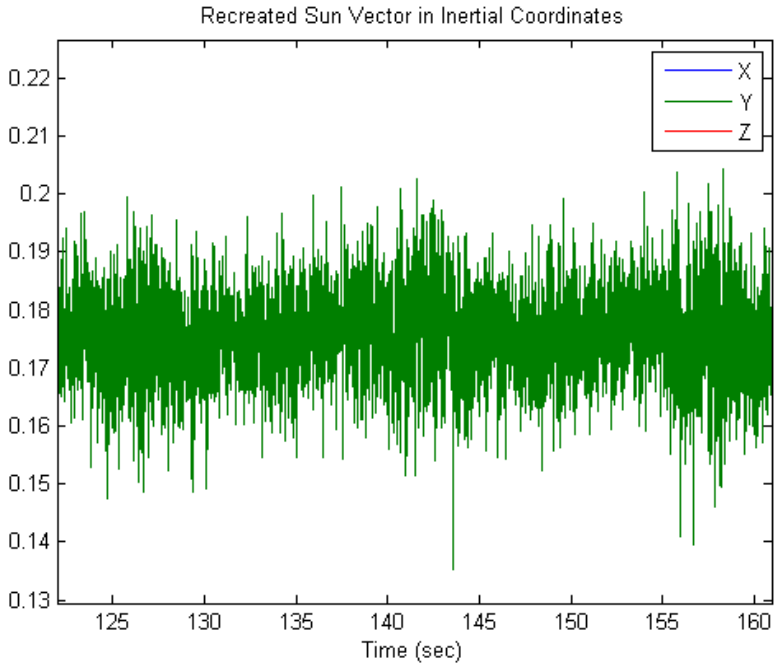

Fig. 5.12: Inertial sun vector Y component using third order calibration.

measurement respectively. The units of the measurements are in meters.

$$\begin{bmatrix} {}^{I}S_{xm} \\ {}^{I}S_{ym} \\ {}^{I}S_{zm} \end{bmatrix} = \begin{bmatrix} 5.63\,m \\ 1.04\,m \\ 0\,m \end{bmatrix} \tag{5.39}$$

5.5.4 Uncertainty Analysis

After the measurements were made the uncertainty analysis could be conducted. It was determined that both measurements on the X and Y components have an uncertainty of 0.05 m, due to the spin axis of the rotation table not being aligned with the floor of the lab the Z component measurement has an uncertainty of 0.2 m. The subscript n stands for normalized.

$${}^{I}S = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{5.40}$$

$$X = 5.63\,m \pm 0.05\,m \tag{5.41}$$

$$Y = 1.04\,m \pm 0.05\,m \tag{5.42}$$

$$Z = 0\,m \pm 0.2\,m \tag{5.43}$$

$$s(X,Y) = \sqrt{X^2 + Y^2 + Z^2} \tag{5.44}$$

$$s_0 = 5.72\,m \tag{5.45}$$

$$^{I}S_n = \begin{bmatrix} 0.983 \\ 0.181 \\ 0 \end{bmatrix} \tag{5.46}$$

Above is the initial setup of the uncertainty analysis. Now the u's and the partials of the u's can be found.

$$u_x = \frac{X}{s(X,Y,Z)} \tag{5.47}$$

$$u_y = \frac{Y}{s(X,Y,Z)} \tag{5.48}$$

$$u_z = \frac{Z}{s(X,Y,Z)} \tag{5.49}$$

First u_x is found.

$$u_x = u_{x_0} + \frac{\partial u_x}{\partial X}\delta X + \frac{\partial u_x}{\partial Y}\delta Y + \frac{\partial u_x}{\partial Z}\delta Z \tag{5.50}$$

$$\frac{\partial u_x}{\partial X} = \frac{\partial}{\partial X}\left(\frac{X}{s}\right) = \frac{1}{s} - \frac{X}{(s)^2}\frac{\partial s}{\partial X} \tag{5.51}$$

$$\frac{\partial u_x}{\partial Y} = \frac{\partial}{\partial Y}\left(\frac{X}{s}\right) = \frac{-X}{(s)^2}\frac{\partial s}{\partial Y} \tag{5.52}$$

$$\frac{\partial u_x}{\partial Z} = \frac{\partial}{\partial Z}\left(\frac{X}{s}\right) = \frac{-X}{(s)^2}\frac{\partial s}{\partial Z} \tag{5.53}$$

$$\left.\frac{\partial s}{\partial X}\right|_{(X_0,Y_0,Z_0)} = \frac{\partial}{\partial X}(X^2 + Y^2 + Z^2)^{1/2}|_{(X_0,Y_0,Z_0)} = \frac{X_0}{s_0} \tag{5.54}$$

$$\left.\frac{\partial s}{\partial Y}\right|_{(X_0,Y_0,Z_0)} = \frac{Y_0}{s_0} \tag{5.55}$$

$$\left.\frac{\partial s}{\partial Z}\right|_{(X_0,Y_0,Z_0)} = \frac{Z_0}{s_0} \tag{5.56}$$

$$u_x = u_{x_0} + \left[\frac{1}{s_0} - \frac{X_0^2}{s_0^3}\right]\delta X + \left[\frac{-X_0 Y_0}{s_0^3}\right]\delta Y + \left[\frac{-X_0 Z_0}{s_0^3}\right]\delta Z \tag{5.57}$$

$$u_x = 0.98 + \left[\frac{1}{5.73} - \frac{5.63^2}{5.73^3}\right]0.05 + \left[\frac{-5.63 * 1.04}{5.73^3}\right]0.05 + 0 \tag{5.58}$$

$$u_x = 0.983 \pm 0.0013 \tag{5.59}$$

Then u_y is found.

$$u_y = u_{y_0} + \frac{\partial u_y}{\partial X}\delta X + \frac{\partial u_y}{\partial Y}\delta Y + \frac{\partial u_y}{\partial Z}\delta Z \tag{5.60}$$

$$\frac{\partial u_y}{\partial X} = \frac{\partial}{\partial X}\left(\frac{Y}{s}\right) = \frac{-Y}{(s)^2}\frac{\partial s}{\partial X} \tag{5.61}$$

$$\frac{\partial u_y}{\partial Y} = \frac{\partial}{\partial Y}\left(\frac{Y}{s}\right) = \frac{1}{s} - \frac{Y}{(s)^2}\frac{\partial s}{\partial Y} \tag{5.62}$$

$$\frac{\partial u_y}{\partial Z} = \frac{\partial}{\partial Z}\left(\frac{Y}{s}\right) = \frac{-Y}{(s)^2}\frac{\partial s}{\partial Z} \tag{5.63}$$

$$u_y = u_{y_0} + \left[\frac{-Y_0 X_0}{s_0^3}\right]\delta X + \left[\frac{1}{s_0} - \frac{Y_0^2}{s_0^3}\right]\delta Y + \left[\frac{-Y_0 Z_0}{s_0^3}\right]\delta Z \tag{5.64}$$

$$u_y = 0.181 + \left[\frac{-1.04 * 5.63}{5.73^3}\right]0.05 + \left[\frac{1}{5.73} - \frac{1.04^2}{5.73^3}\right]0.05 + 0 \tag{5.65}$$

$$u_y = 0.181 \pm 0.0069 \tag{5.66}$$

Finally u_z is found.

$$u_z = u_{z_0} + \frac{\partial u_z}{\partial X}\delta X + \frac{\partial u_z}{\partial Y}\delta Y + \frac{\partial u_z}{\partial Z}\delta Z \tag{5.67}$$

$$\frac{\partial u_z}{\partial X} = \frac{\partial}{\partial X}\left(\frac{Z}{s}\right) = \frac{-Z}{(s)^2}\frac{\partial s}{\partial X} \tag{5.68}$$

$$\frac{\partial u_z}{\partial Y} = \frac{\partial}{\partial Y}\left(\frac{Z}{s}\right) = \frac{-Z}{(s)^2}\frac{\partial s}{\partial Y} \tag{5.69}$$

$$\frac{\partial u_z}{\partial Z} = \frac{\partial}{\partial Z}\left(\frac{Z}{s}\right) = \frac{1}{s} - \frac{Z}{(s)^2}\frac{\partial s}{\partial Z} \tag{5.70}$$

$$u_z = u_{z_0} + \left[\frac{-Z_0 X_0}{s_0^3}\right]\delta X + \left[\frac{-Z_0 Y_0}{s_0^3}\right]\delta X + \left[\frac{1}{s_0} - \frac{Z_0^2}{s_0^3}\right]\delta Z \tag{5.71}$$

$$u_z = 0 + 0 + 0 + \left[\frac{1}{5.73} - \frac{0^2}{5.73^3}\right]0.2 \tag{5.72}$$

$$u_z = 0 \pm 0.035$$

It can be seen in Table 5.2 that the X, Y, and Z components of the inertial sun vector found using the least squares solution fall within the error bounds found for the uncertainty analysis. This gives more confidence in results for the sun sensor calibration.

5.5.5 Azimuth

Since the sun vector in body coordinates is known, the azimuth can be found by using simple trigonometry. The vector to the sun in body coordinates is represented with ${}^bS$, and ${}^bS_x$, ${}^bS_y$, *and* ${}^bS_z$ are the individual components of that vector. The azimuth can be found by $az = tan^{-1}(\frac{{}^bS_y}{{}^bS_x})$. The azimuth is the rotation of the mock satellite if there is only rotation about the z axis.

Results

Figure 5.13 shows the plot of the raw sun sensor voltages. Figure 5.14 shows that the raw voltages for one pass.

The calibration is then used to convert the voltages from the sun sensor into the vector to the sun in the sun sensor coordinates. The vector to the sun is then rotated into body coordinates. Figure 5.15 is a plot of the vector components of the sun vector in the sun sensor coordinates for the entire test. In the plot, when all the vector components go to zero that is when the sun sensor does not see the sun. Figure 5.16 is the same plot as Figure 5.15 but only one pass is shown.

Figure 5.17 is the plot of the sun vector in body coordinates for the entire test. Figure 5.18 is the same plot but it is zoomed in so only one pass is shown. The main difference

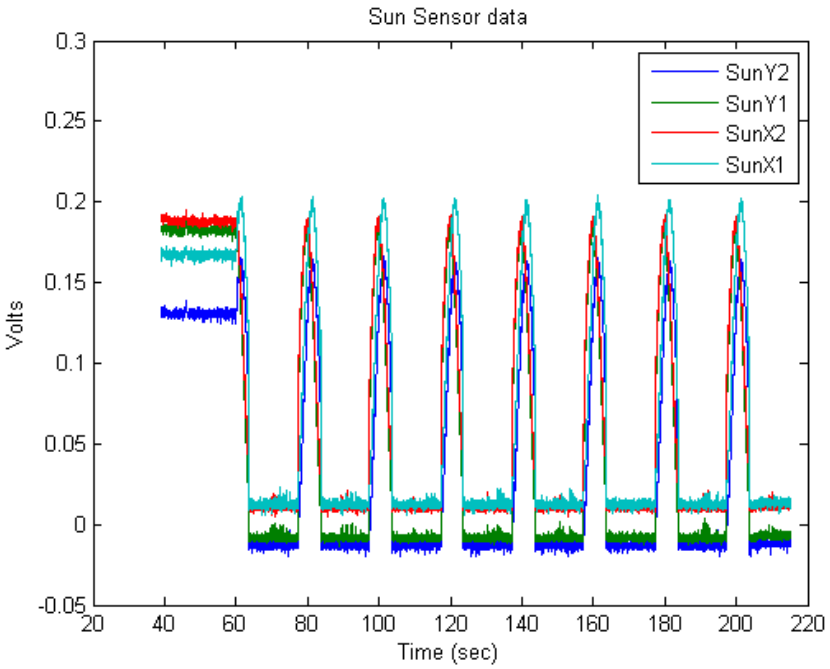

Fig. 5.13: Raw sun sensor voltages for Spin 1 test.

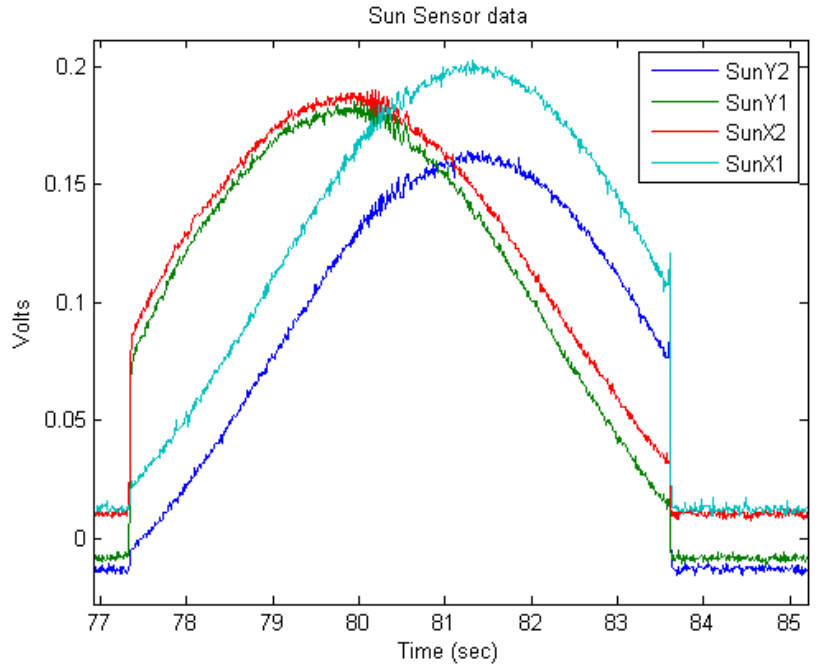

Fig. 5.14: Raw sun sensor voltage for one pass during Spin 1 test.

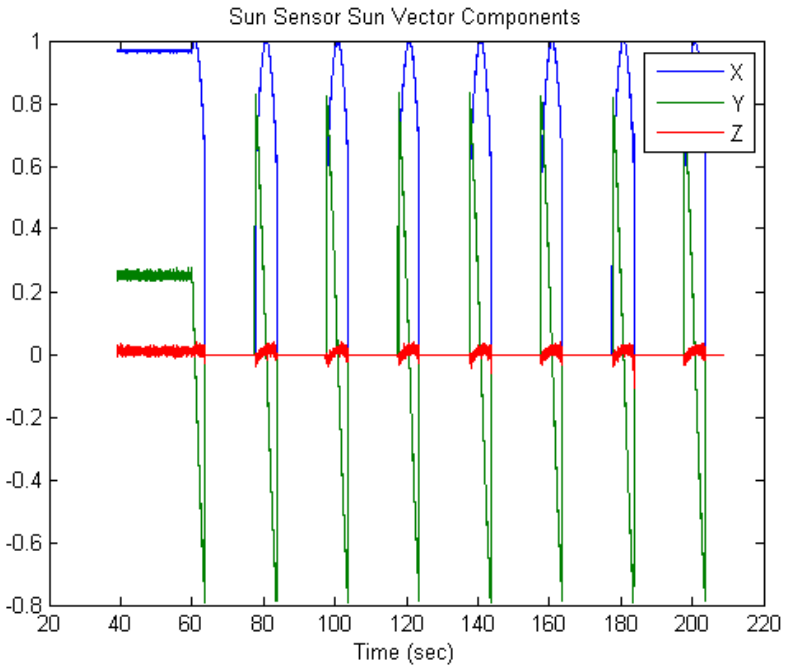

Fig. 5.15: Sun vector in sun sensor coordinates for Spin 1 test.

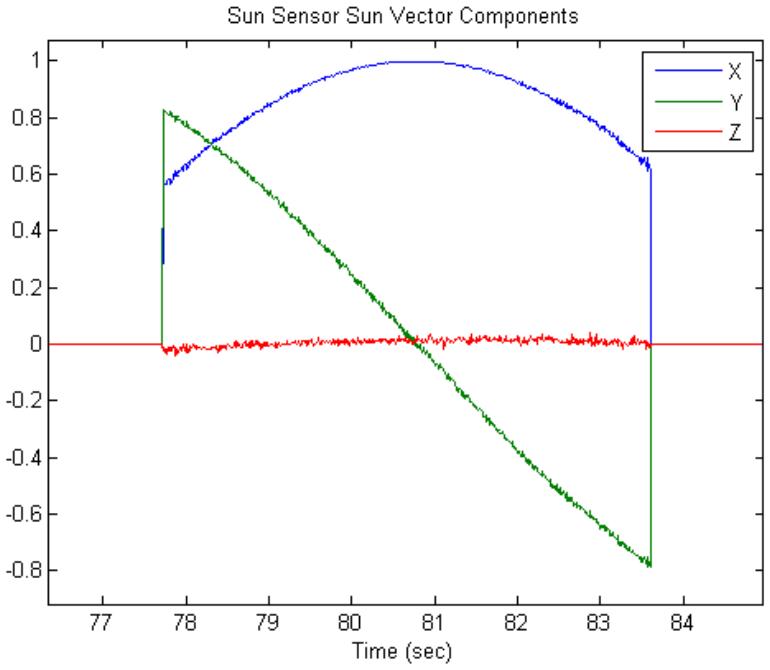

Fig. 5.16: Sun vector in sun sensor coordinates for one pass during Spin 1 test.

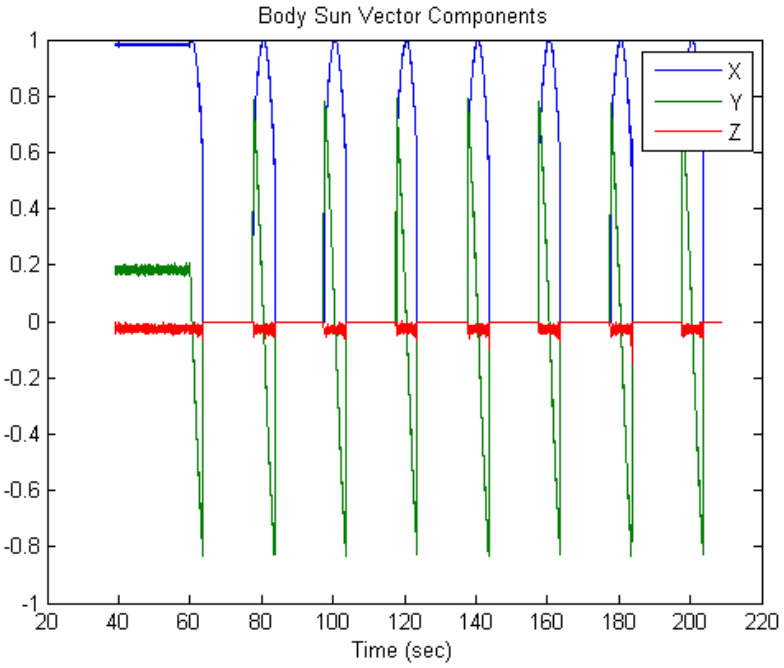

Fig. 5.17: Sun vector in body coordinates for Spin 1 test.

between the sun vector plots in sun sensor and body coordinates is that the Z component of the vector when it is in body coordinates, is more constant. This is evidence that the mounting alignment angles that were found for the sun sensor are correct, because when the Z component is constant there is only a rotation about the Z axis.

Figure 5.19 is the azimuth found for the entire test compared with the rotation found by the encoder. Figure 5.20 is the same as Figure 5.19 but zoomed in so only one pass of the sun is shown. In the plot the sun sensor is able to get results for about 100° of a revolution. Both of these plots show that the azimuth is pretty close to the rotation angle found by the encoder.

The error found in the sun sensor is shown in Figure 5.21. The rms error of the azimuth is 0.5°, but this is only for the one axis approach. Figure 5.22 shows the error in the azimuth from the sun sensor for one pass. It can be seen in Figure 5.22 that the sun sensor accuracy degrades as it approaches the limits of it field of view.

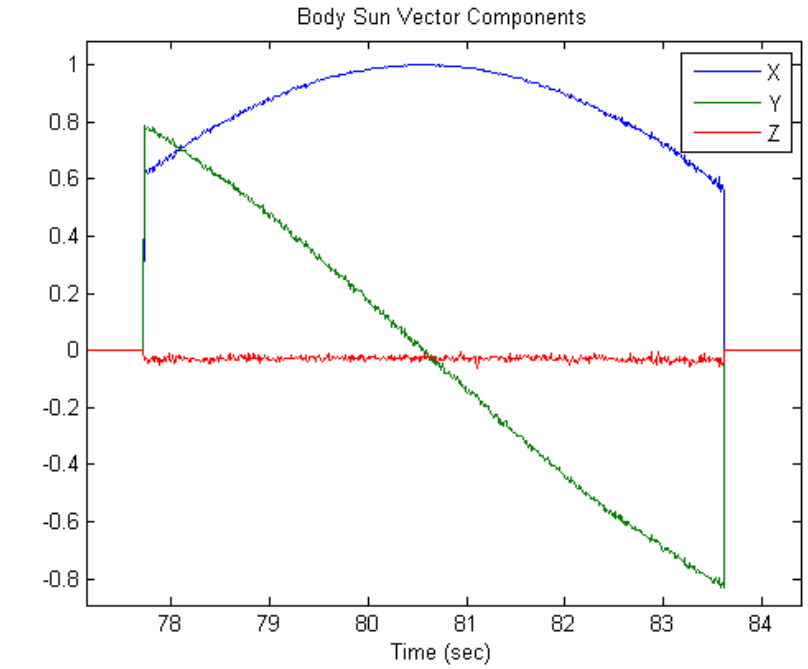

Fig. 5.18: Sun vector in body coordinates for one pass during Spin 1 test.

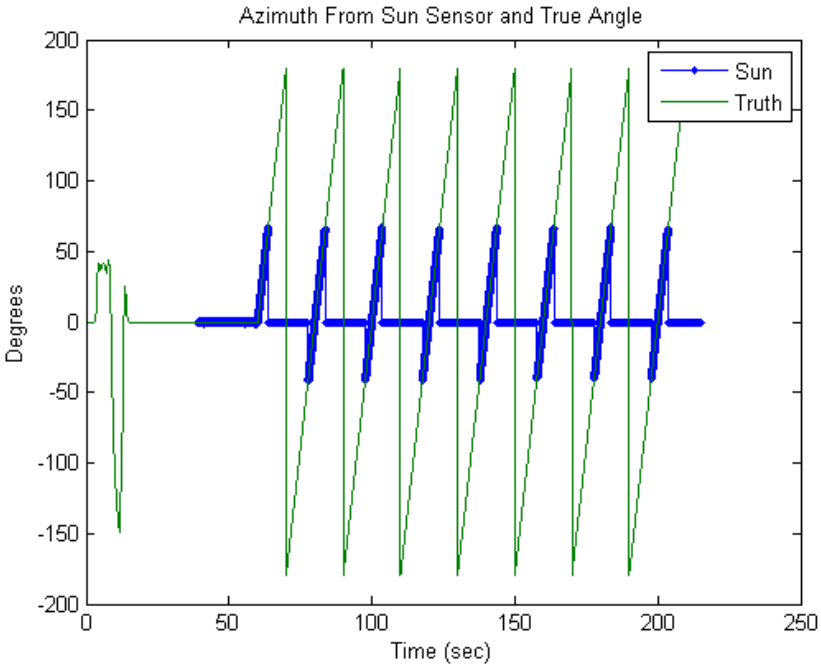

Fig. 5.19: Azimuth from sun sensor and the encoder angle for Spin 1 test.

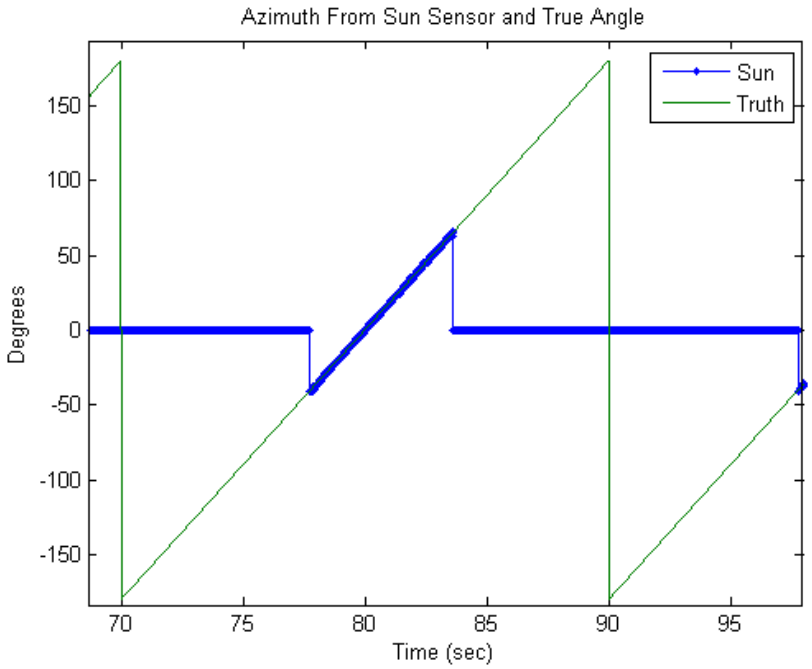

Fig. 5.20: Azimuth from sun and the encoder angle for only one pass during Spin 1 test.

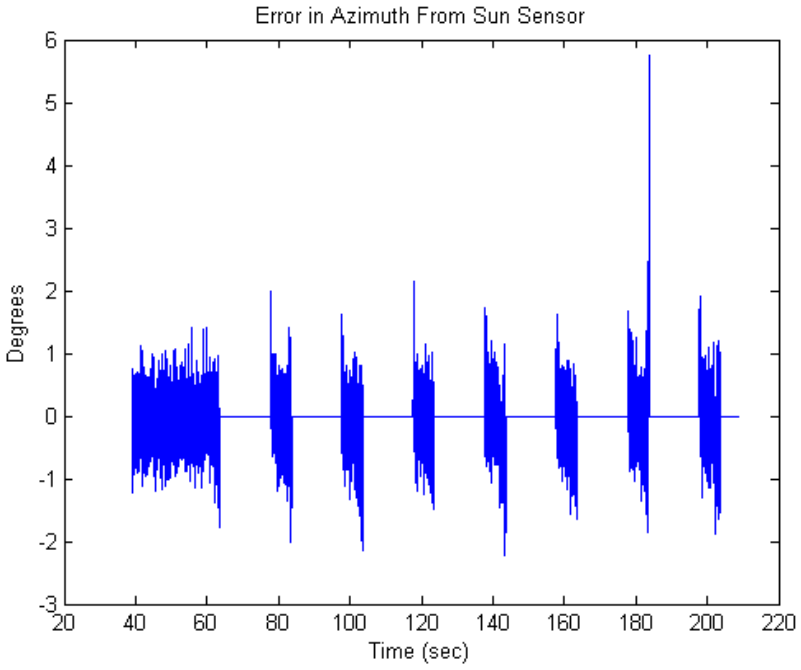

Fig. 5.21: Azimuth error for Spin 1 test.

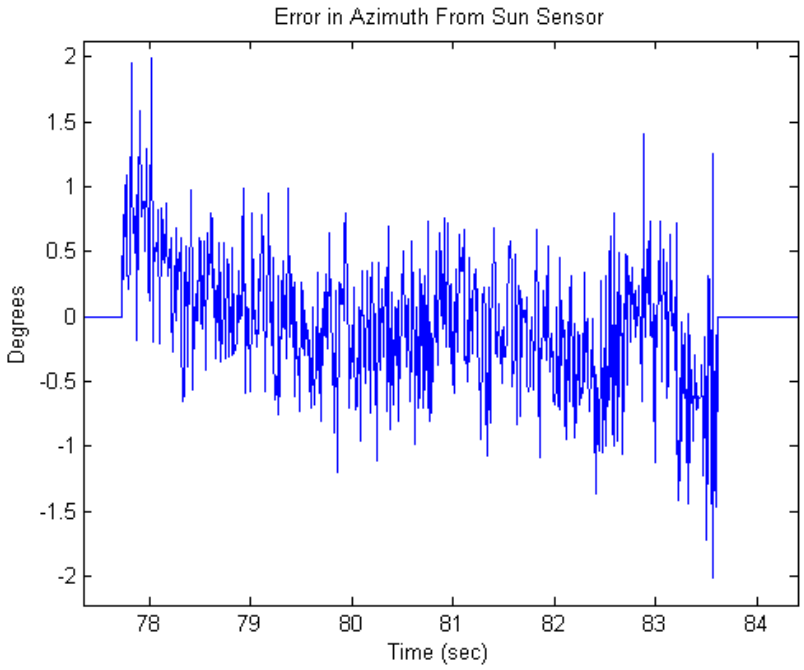

Fig. 5.22: Azimuth error for one pass during Spin 1 test.

Chapter 6

Attitude Determination Algorithms

6.1 Triad

The TRIAD method is fairly straight forward, the process that is explained in Subsection 2.4.1 is used to find the results using the TRIAD method.

6.2 Q-Method

The quaternion is found by using Davenport's q-method. When using Davenport's q-method the K matrix is first found. After K is found, the quaternion is found by finding the eigenvalues and eigenvectors of K. The estimated quaternion is the eigenvector that corresponds with the maximum eigenvalue. The estimated attitude is then compared to the true attitude to find the accuracy of the ADS.

6.2.1 K Matrix

To find the K matrix first, the D matrix is found. The implicit equation for the D matrix is shown in the following equation. In the equation σ is the weighting matrix, $\tilde{b}$ is the measured vector, and r is the known vector.

$$D = \sum_{j=1}^{N} \sigma_j^{-2} \tilde{b}_j r_j^T \tag{6.1}$$

The explicit form of this equation for this problem is shown in the following equation. Both vectors are weighted the same for this problem, which means σ can be ignored.

$$D =^B B *^I B^T +^B S *^I S^T \tag{6.2}$$

D is a 3x3 matrix.

$$c = trace(D) \tag{6.3}$$

c is a scalar.

$$S = D + D^T \tag{6.4}$$

S is a 3x3 matrix.

$$Z = \begin{bmatrix} D_{2,3} - D_{3,2} \\ D_{3,1} - D_{1,3} \\ D_{1,2} - D_{2,1} \end{bmatrix} \tag{6.5}$$

Z is a 3x1 vector.

$$K = \begin{bmatrix} S - c * I_{3x3} & Z \\ Z^T & c \end{bmatrix} \tag{6.6}$$

K is a 4x4 matrix.

6.2.2 Estimated Quaternion

As stated previously, to find the optimal quaternion, the eigenvalues and eigenvectors need to be found. K is a 4x4 matrix, and so there will be four eigenvalues and four eigenvectors. The optimal quaternion is the eigenvector that corresponds to the maximum eigenvalue. The eigenvalues and eigenvectors are found by solving the following equation.

$$K\hat{q} = \lambda\hat{q} \tag{6.7}$$

The optimal quaternion found is the estimated quaternion for the ADS which contains the estimated attitude of the mock satellite.

6.3 Kalman Filter

The Kalman filter used in this experimental thesis was originally built by Jandak [18]. The filter was then modified by Fullmer [32]. The Kalman filter was only used on the spin

tests. The process model for the rotational stage assumed independent angular acceleration about each axis driven only by the process noise. The process model is shown in Equations 6.8 through 6.11.

$$\ddot{\phi} = \nu_x \; \dot{\phi}(0) = \phi(0) = 0 \tag{6.8}$$

$$\ddot{\theta} = \nu_y \; \dot{\theta}(0) = \theta(0) = 0 \tag{6.9}$$

$$\ddot{\psi} = \nu_z \; \dot{\psi}(0) = \omega_0 \; \psi(0) = \psi_0 \tag{6.10}$$

$$\dot{\omega} = -\omega \times I\omega \tag{6.11}$$

The standard deviation of the process noise about the rotation axis was estimated by calculating the residual value from a discrete time double integrator using actual encoder data as approximately 1e-5 $\frac{rad}{s^2}$. This value was used for all three axes. The initial state was defined using encoder data. The software implementing the filter was validated by using the encoder data to rotate the inertial sun and magnetometer vectors into body coordinates and then adding white noise. The errors from this model agreed well with the predicted values from the propagated covariance.

Chapter 7

Results

7.1 TRIAD Results

Only the results for Spin 2 using the TRIAD method are presented. The Euler angles found using the TRIAD method are shown in Figures 7.1 and 7.2. The total error in the estimated attitude found using the TRIAD method is shown in Figure 7.3.

7.2 Q-Method Results

The results of six different tests using the q-method are presented. Table 7.1 presents the error found for all of the different tests. As can be seen, the accuracy of the attitude solution is much more accurate when the magnetic field and sun vector are closer to being orthogonal. As explained in Chapter 2 the attitude solution using the q-method falls apart when the vector measurements are parallel.

It would be redundant to show all of the results for every test. All of the results are shown for Spin 1 and some results are shown for Spin 3, Sweep 1, and Sweep 2.

7.2.1 Spin 1

Table 7.1: Results to The Six Different Tests

Test Name	Angle Between Magnetic Field Vector and Sun Vector	Total Attitude Error
Spin 1	99°	0.53°
Spin 2	99°	0.54°
Spin 3	12°	2.22°
Spin 4	12°	2.42°
Sweep 1	12°	2.18°
Sweep 2	93°	0.52°

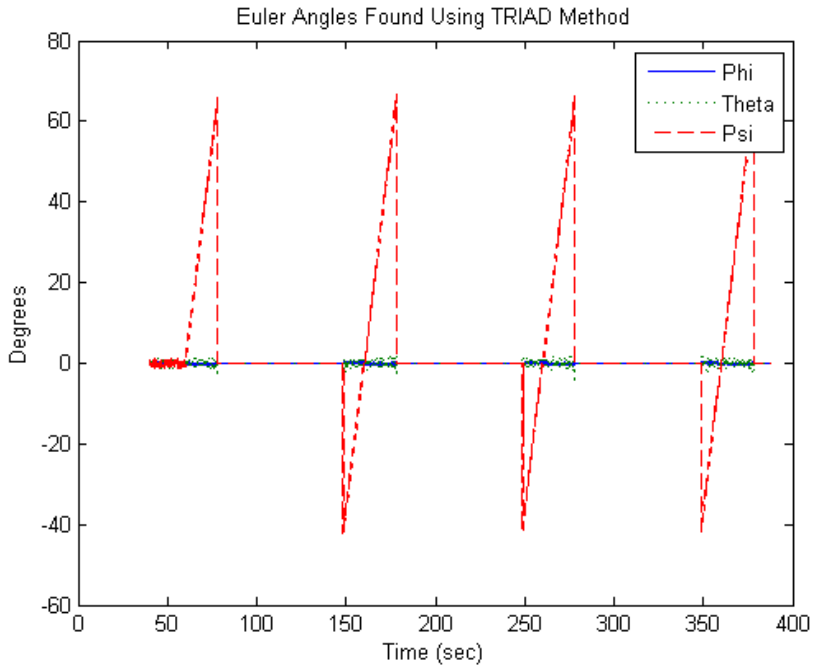

Fig. 7.1: Euler angles found for Spin 2 using the TRIAD method.

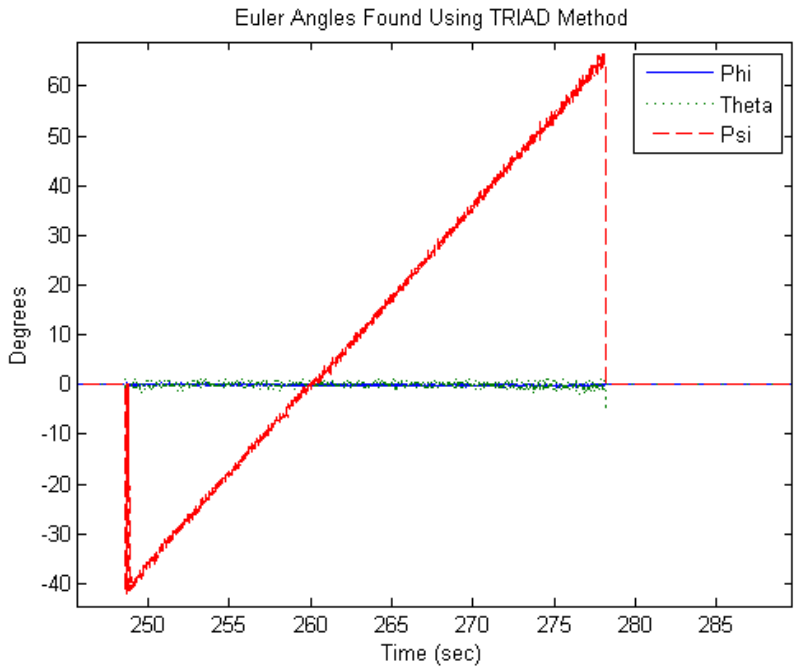

Fig. 7.2: Euler angles found for Spin 2 using the TRIAD method only one pass shown.

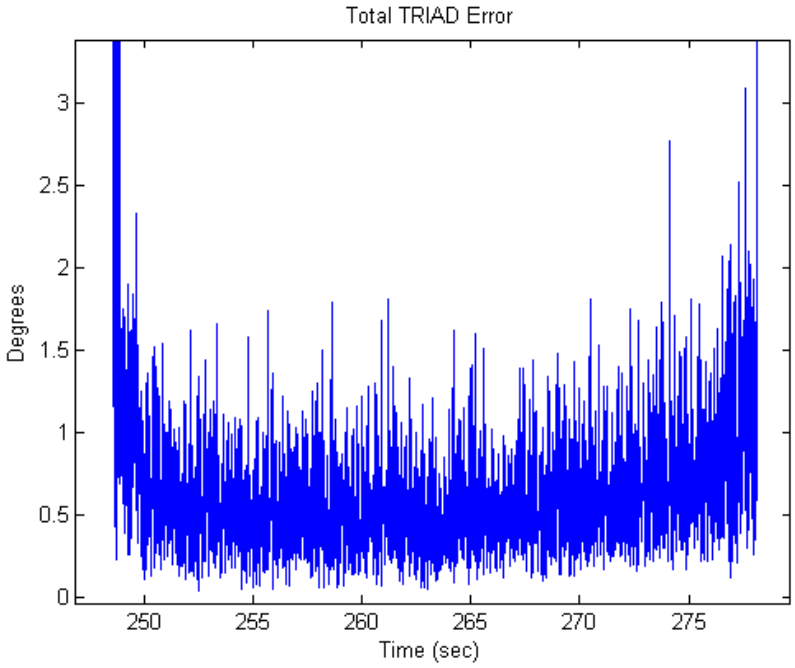

Fig. 7.3: Total error for the TRIAD method for one pass.

Time Alignment

The first thing completed when the data is being analyzed is the alignment of the inertial and body time vectors. Figure 7.4 shows the flashes recorded by both the inertial and body computers for the entire test. Figure 7.5 shows only 10 seconds of the same plot shown in Figure 7.4, this makes it possible to see that the strobe flashes are aligned.

Attitude Results

The quaternion rotation that was found was converted to Euler angles. The Euler angles shown in Figure 7.6 and 7.7. The q-method falls apart when the Sun cannot be seen. The spacecraft needs to have a Kalman filter in order to still have attitude knowledge when the Sun is not in view.

The total error in the q-method is 0.53°, the difference between the encoder ψ and the q-method ψ is 0.33°. Figures 7.8 and 7.9 plot the psi found using the q method with the angle of rotation from the rotation table. Figure 7.10 is the total error in the attitude found using the q method.

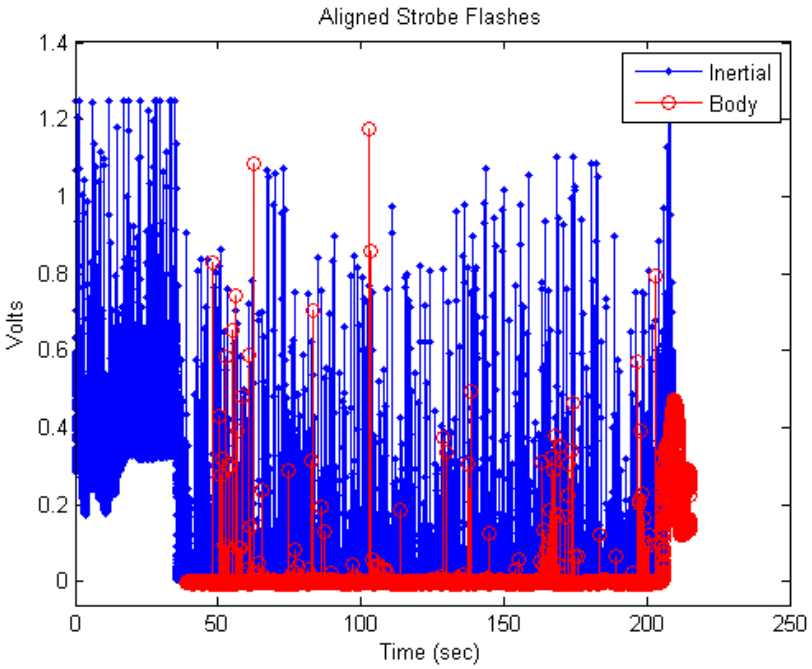

Fig. 7.4: Aligned strobe flashes for entire plot.

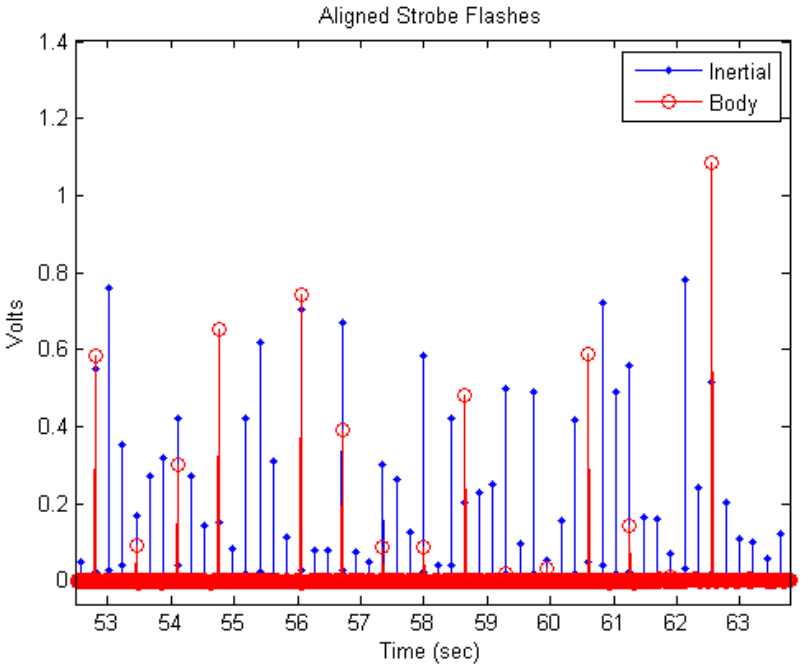

Fig. 7.5: Aligned strobe flashes zoomed in.

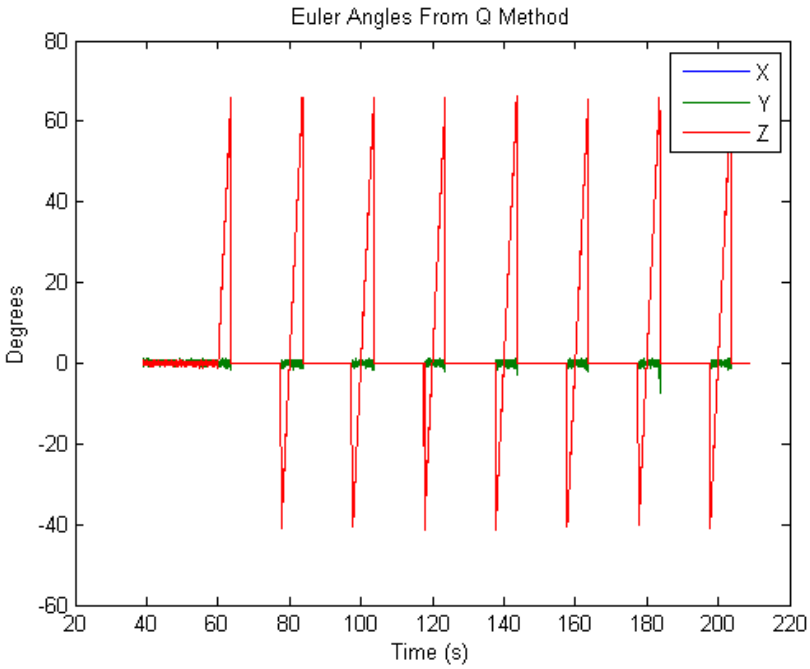

Fig. 7.6: Euler angles from the q-method.

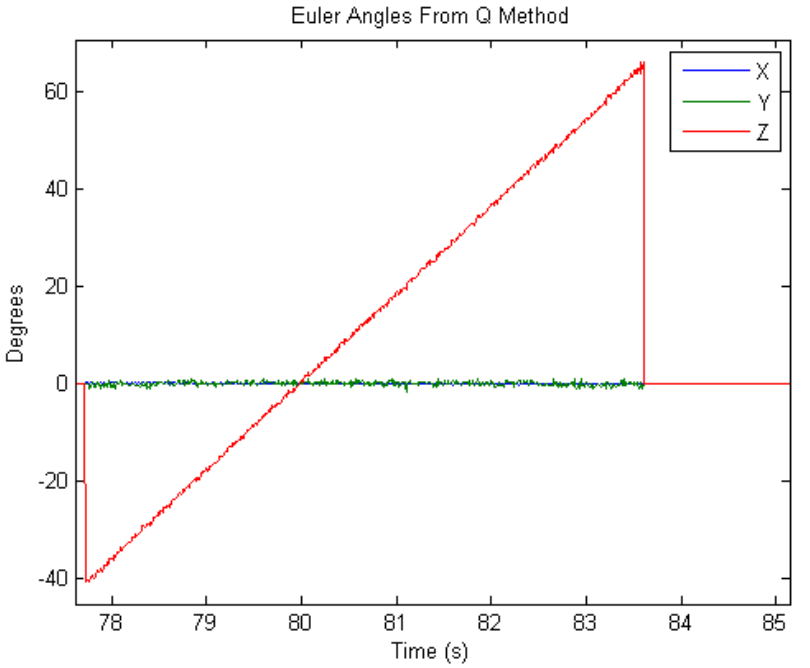

Fig. 7.7: Euler angles from the q-method for on pass.

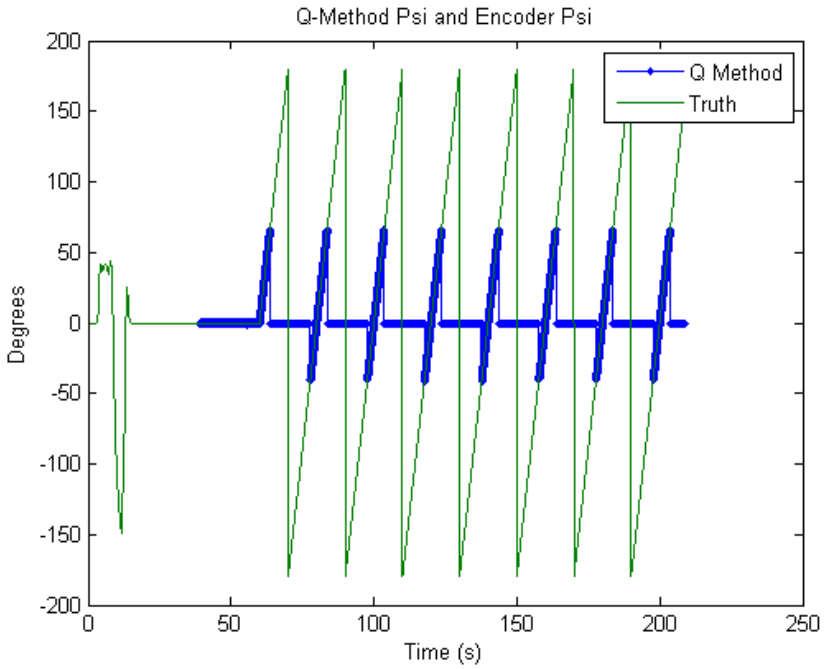

Fig. 7.8: Encoder psi and the q-method psi.

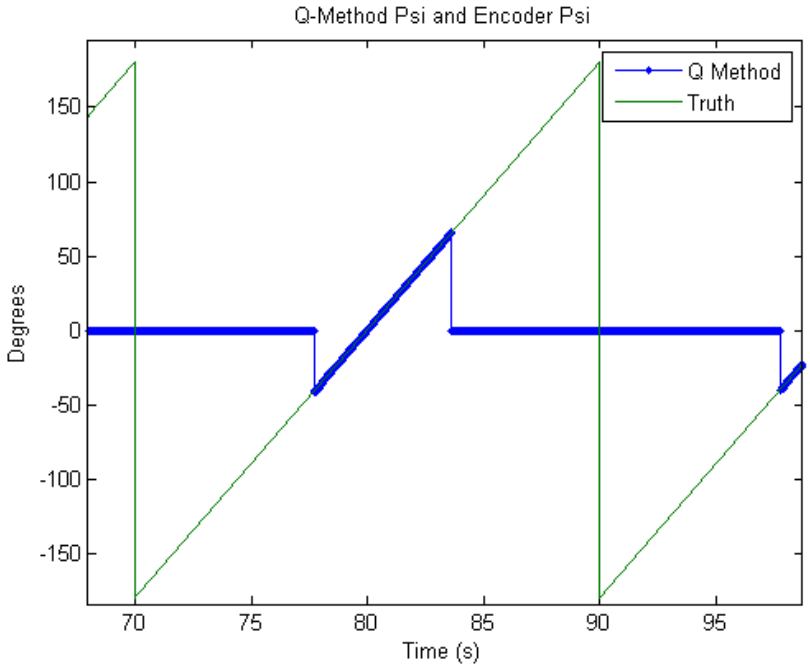

Fig. 7.9: Encoder psi and the q-method psi for one pass.

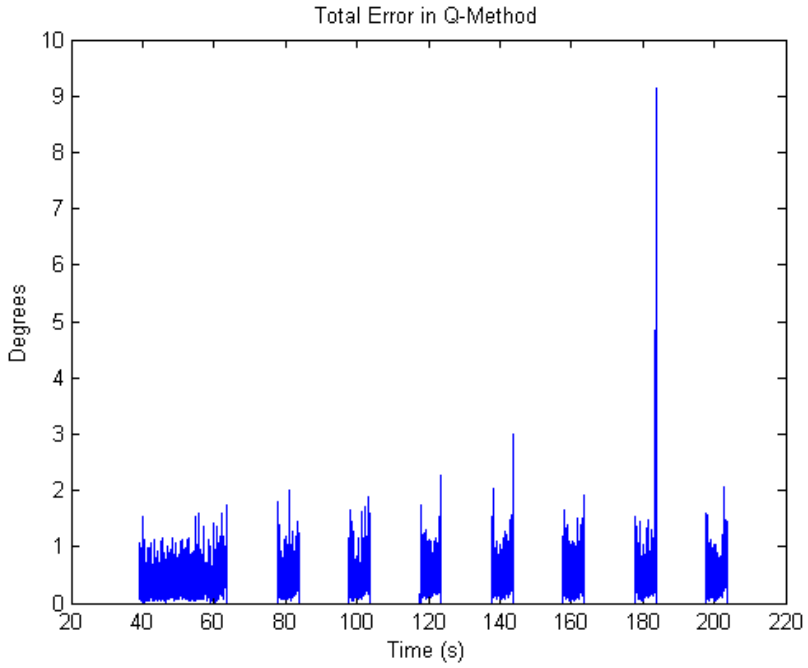

Fig. 7.10: Q-Method total error.

7.2.2 Spin 3, Sweep 1, and Sweep 2

The Euler angles that were found using the q-method for Spin 3, Sweep 1, and Sweep 2 are shown in Figures 7.11 though 7.16. There is a plot of the entire test followed by a zoomed-in plot for each test. In these plots it can be seen that the ϕ found for Spin 3 and Sweep 1 has a lot more noise than the ϕ found for Spin 1 and Sweep 2. This increase in the noise is because the magnetic field vector and the vector to the sun are less orthogonal.

The ψ from the encoder and the q-method are plotted together for Spin 3, Sweep 1, and Sweep 2 in Figures 7.17 through 7.19. It can be seen in the plots that although the total error went up for Spin 3 and Sweep 1, the error in ψ actually went down. The rms error for ψ alone for Spin 3, Sweep 1, and Sweep 2 is 0.25°, 0.24°, and 0.43°, respectively.

The rms error for the attitude found using the q-method for Spin 3, Sweep 1, and Sweep 2 is 2.14°, 2.14°, and 0.54°, respectively. The large difference in the attitude error is due to the orthogonality of the magnetic field and the sun vector. The error of the q-method attitude for Spin 3, Sweep 1, and Sweep 2 are shown in Figures 7.20 through 7.22.

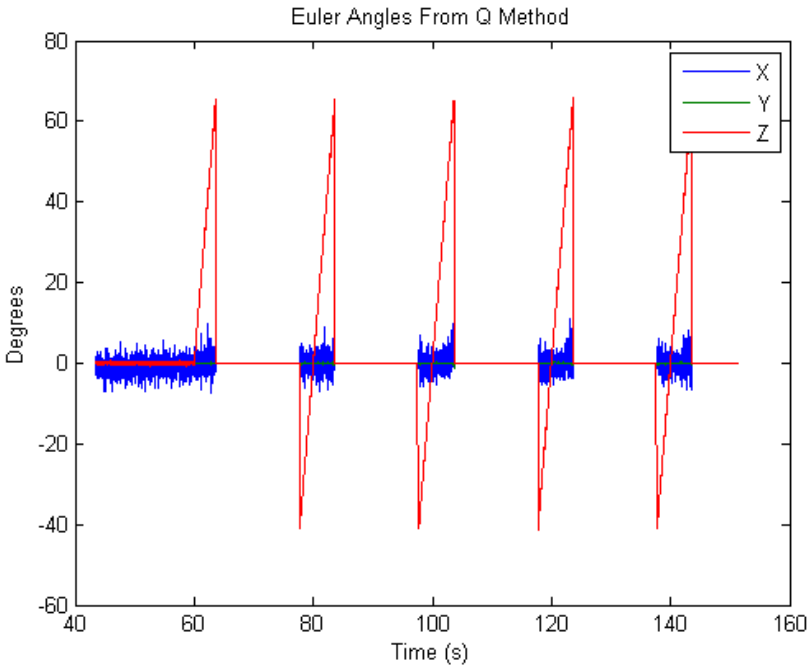

Fig. 7.11: Spin 3 Euler angles found using the q-method.

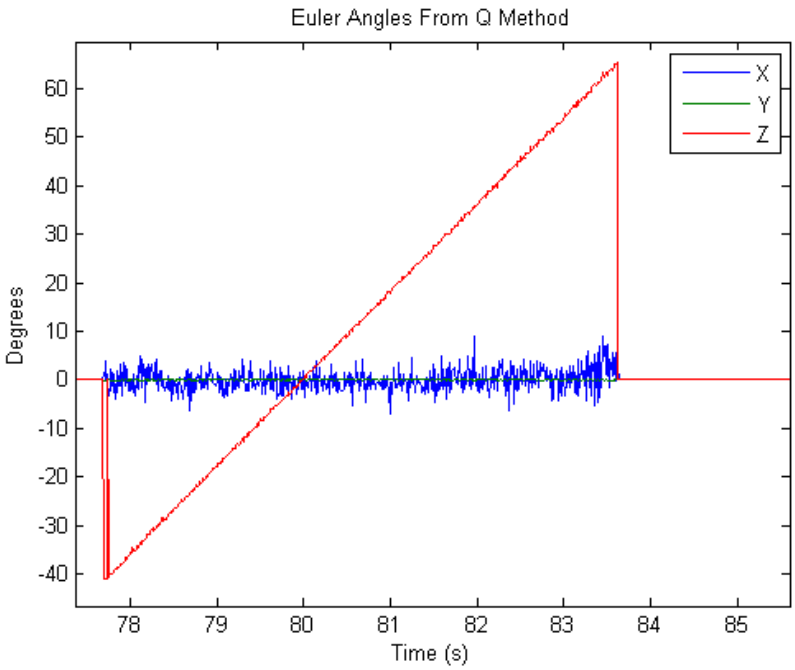

Fig. 7.12: Spin 3 Euler angles found using the q-method for one pass.

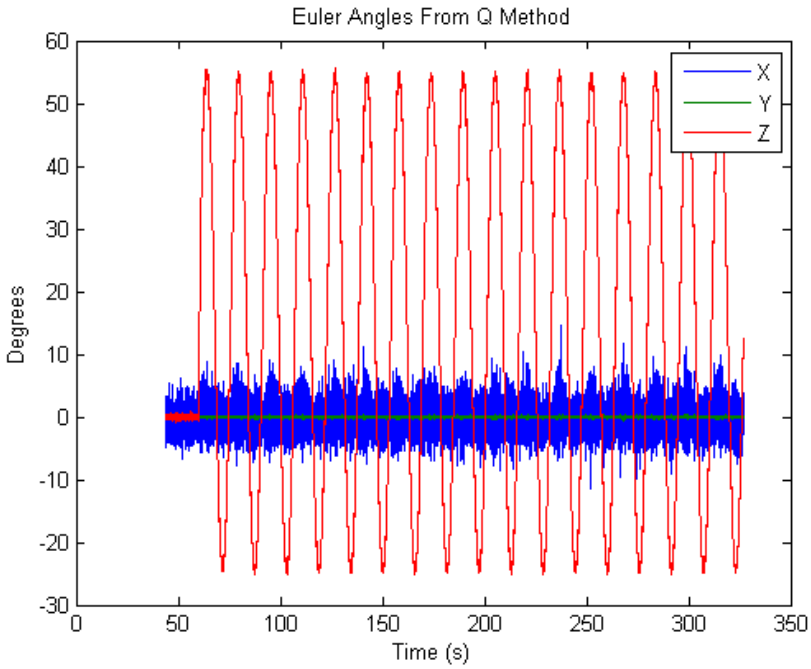

Fig. 7.13: Sweep 1 Euler angles found using the q-method.

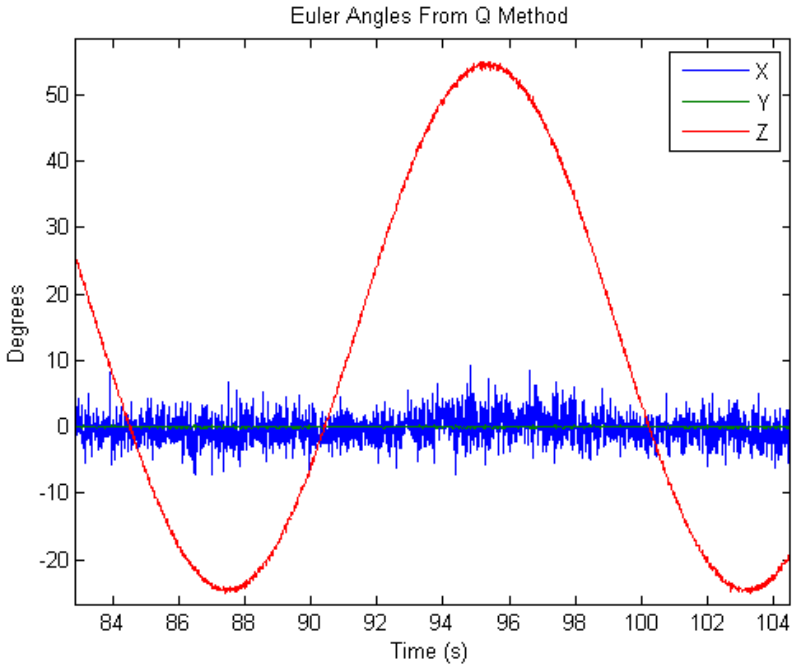

Fig. 7.14: Sweep 1 Euler angles found using the q-method for one oscillation.

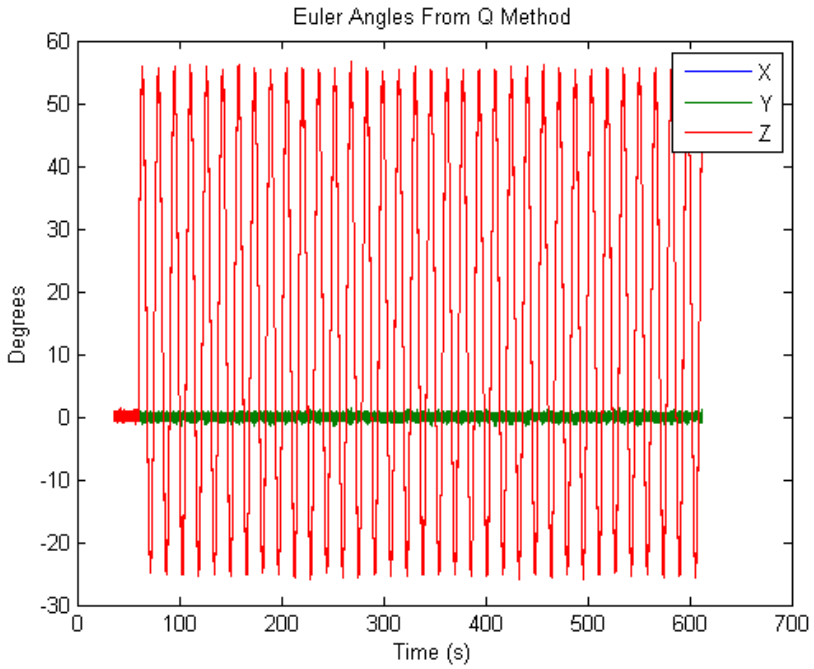

Fig. 7.15: Sweep 2 Euler angles found using the q-method.

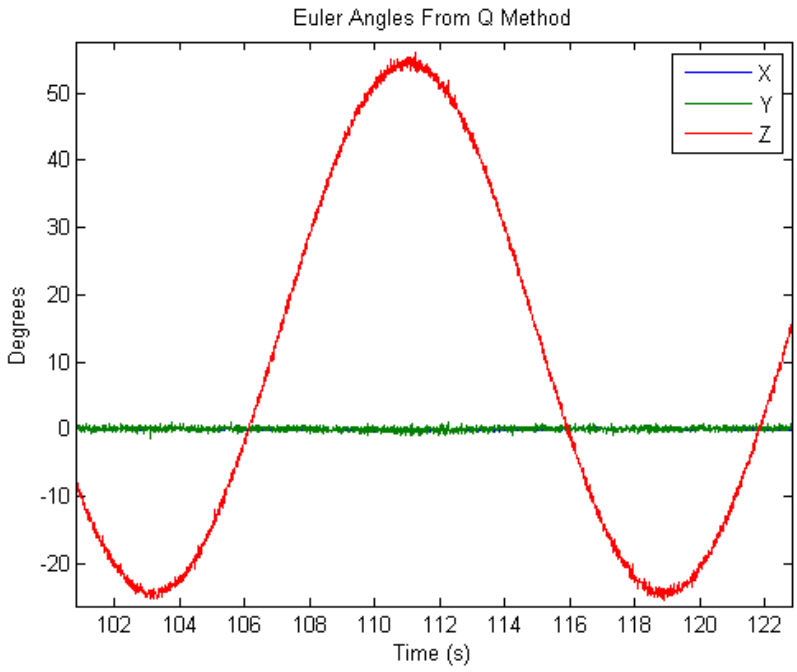

Fig. 7.16: Sweep 2 Euler angles found using the q-method for one oscillation.

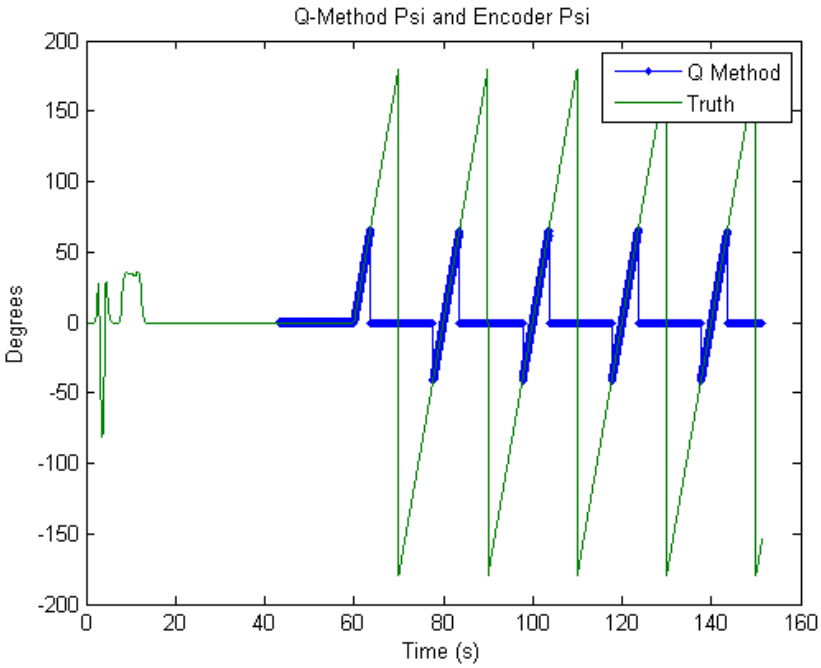

Fig. 7.17: Spin 3 encoder ψ and the q-method ψ.

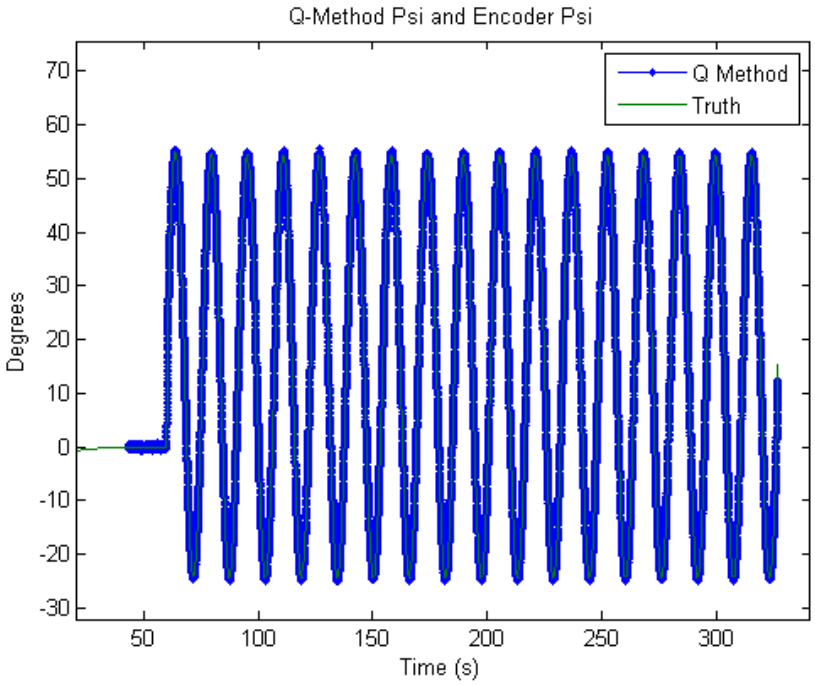

Fig. 7.18: Sweep 1 encoder ψ and the q-method ψ.

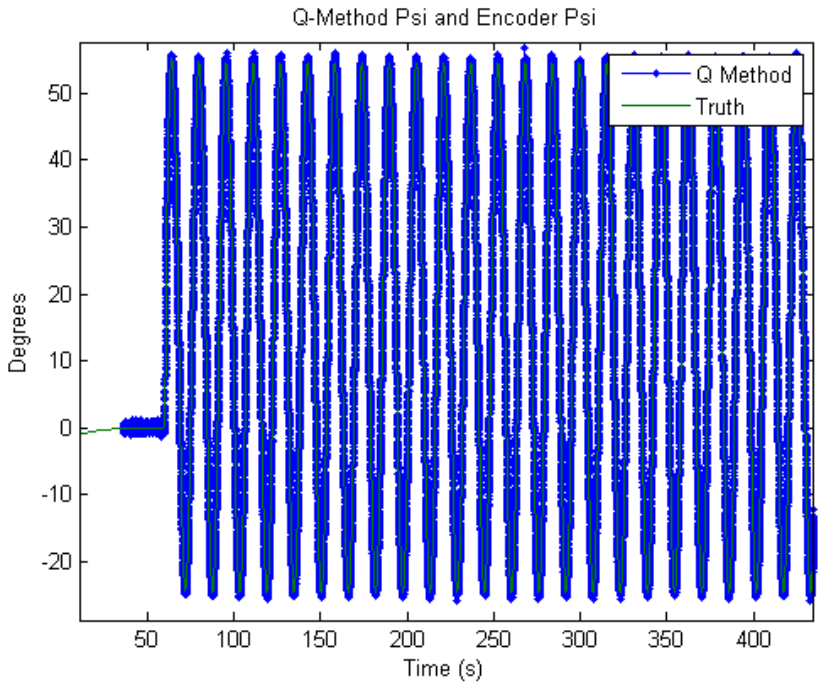

Fig. 7.19: Sweep 2 encoder ψ and the q-method ψ.

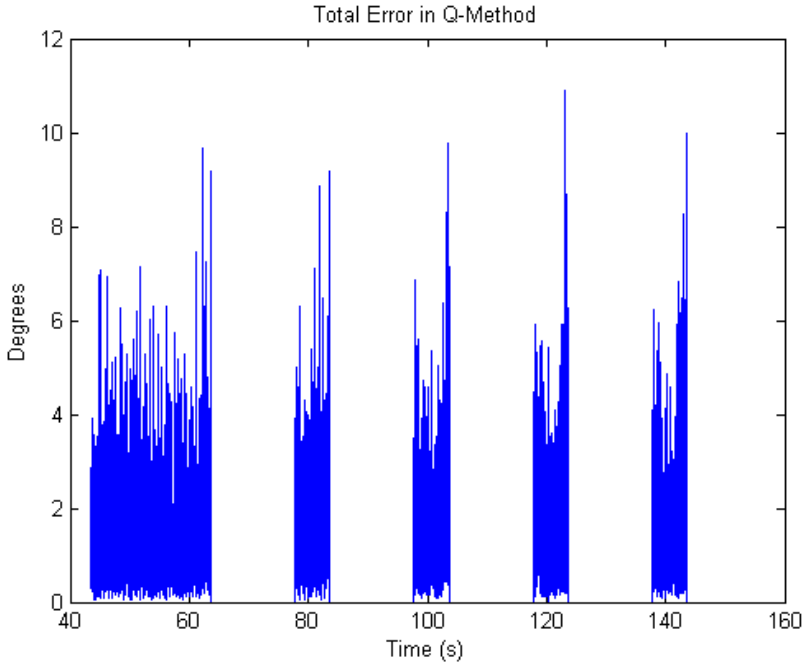

Fig. 7.20: Spin 3 q-method total error.

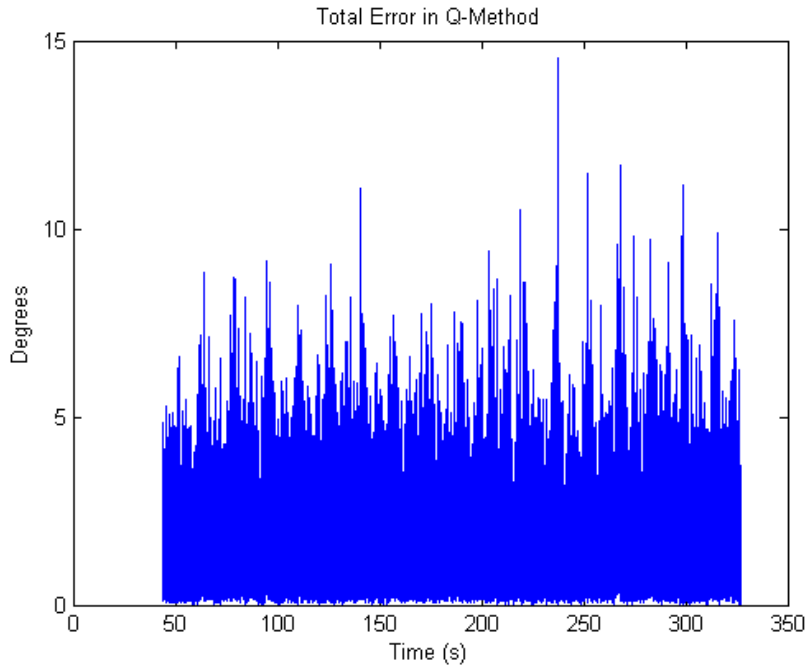

Fig. 7.21: Sweep 1 q-method total error.

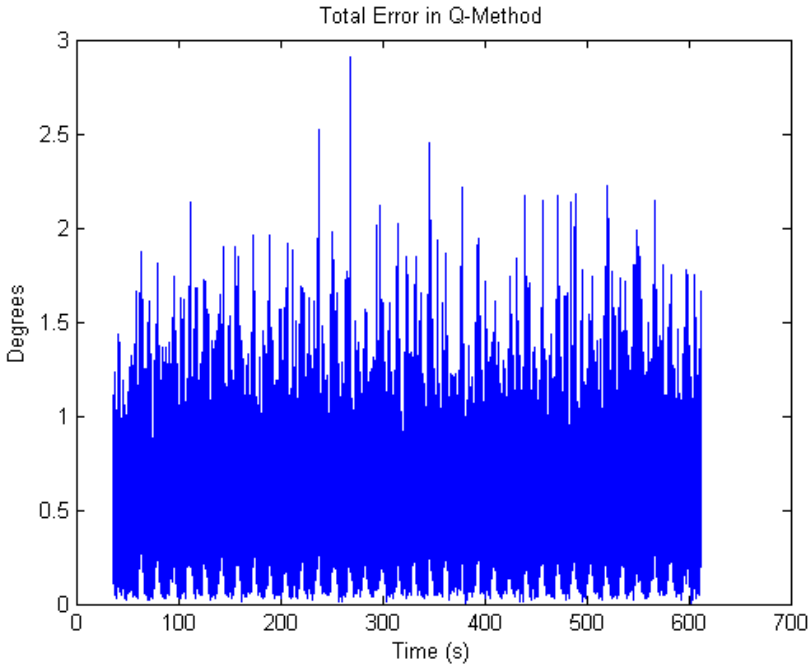

Fig. 7.22: Sweep 2 q-method total error.

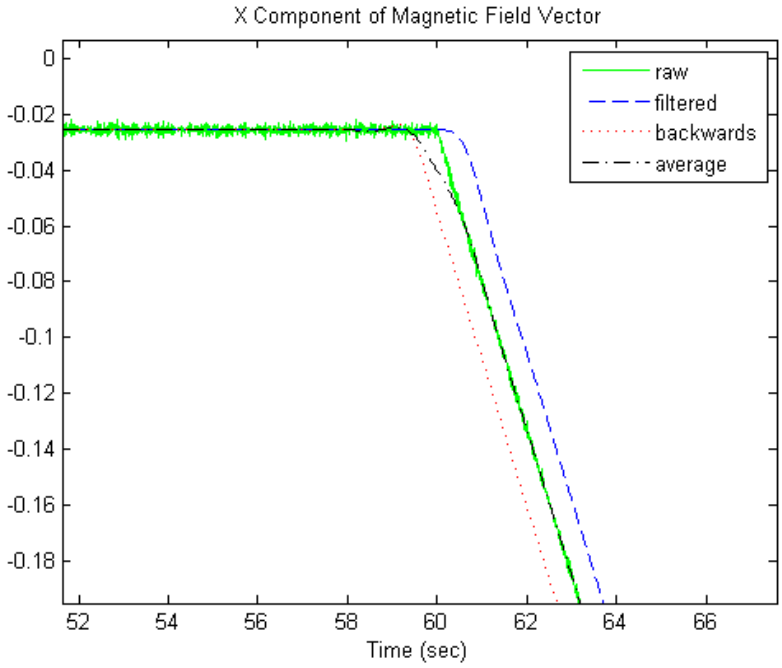

Fig. 7.23: Low pass filtered x component of magnetic field.

Table 7.2: 4th Order Butterworth Low Pass Filter Results

Test	TRIAD		Q Method	
	Error (deg)	Predict (deg)	Error (deg)	Predict (deg)
Spin1	0.48	0.045	0.45	0.033
Spin2	0.27	0.066	0.24	0.048
Spin3	0.53	0.189	0.46	0.187
Spin4	0.3	NA	0.3	NA

7.3 Low Pass Filter

A fourth order Butterworth low pass filter set to $5\frac{rads}{sec}$ was used to filter the data. To compensate for the time lag, the average of the forward and backward filtered data was used. How the filtered magnetometer and sun sensor data line up with the raw data is shown in Figures 7.23 and 7.24. Table 7.2 shows the results found using the fourth order Butterworth filter. In Table 7.2 it can be seen that the TRIAD and q-method results are very close to the same. The plots presented in this section were found using the q method after the raw data was filtered.

Figures 7.25 and 7.26 show the Euler angles found when using the low pass filter. Plots

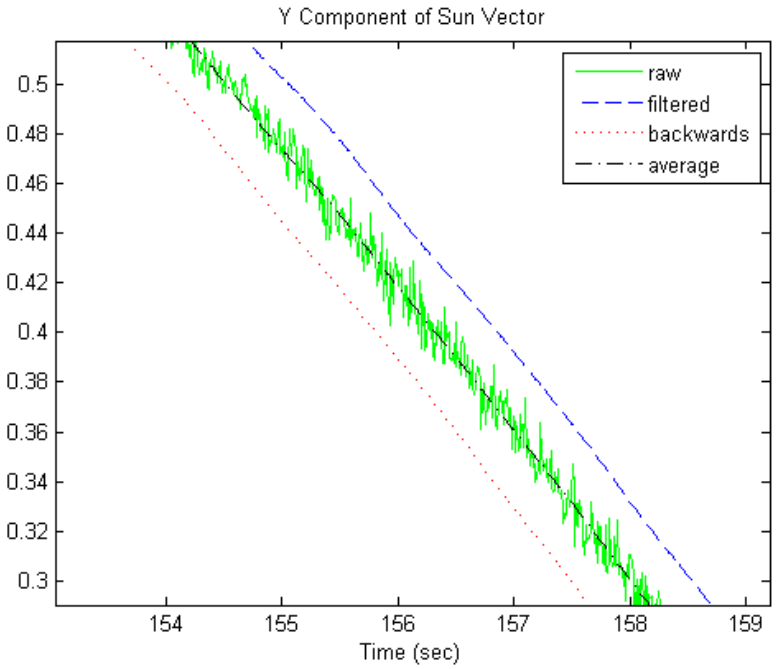

Fig. 7.24: Low pass filtered y component of sun vector.

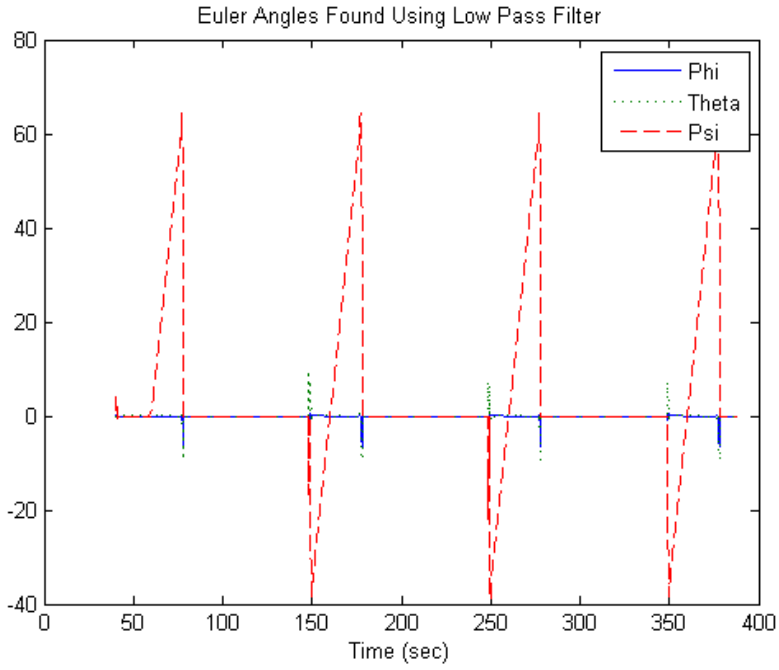

Fig. 7.25: Euler angles found using low pass filter.

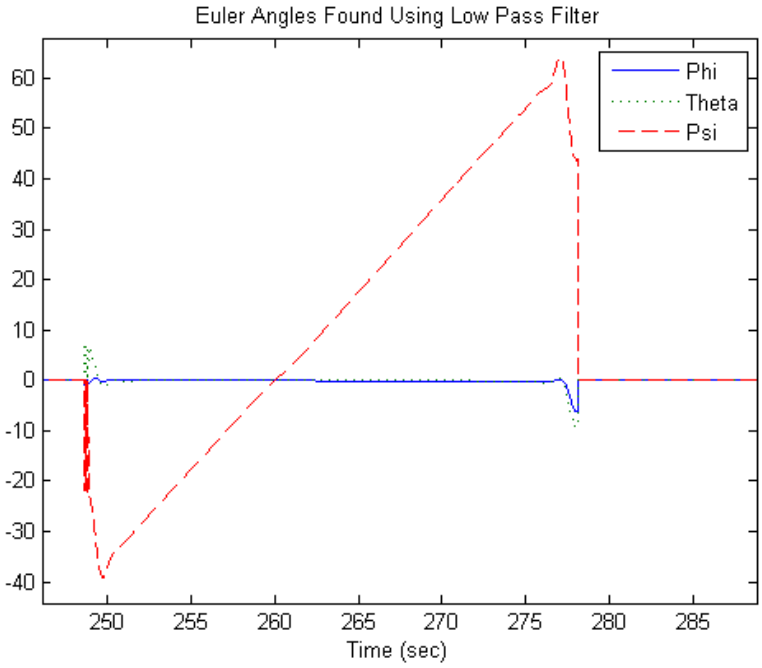

Fig. 7.26: Euler angles found using low pass filter with one pass shown.

of the error in the results found using the fourth order Butterworth low pass filter are shown in Figures 7.27 and 7.28. The spike that occurs at 260 seconds in Figure 7.28 is caused by an anomaly in the encoder. What happens to make this spike appear is that the controller commands the rotation table to go to zero degrees before the rotation table has crossed home and caused the encoder to reset to zero. The controller then commands the rotation table to spin the opposite direction. This causes a large acceleration, but the momentum of the mock spacecraft cause the rotation table to continue spinning till the encoder has reset. The rotation table then continues to rotate as usual. This anomaly is only affects the results found by the different filters.

7.4 Kalman Filter Results

The noise on both the magnetometer and the sun sensor were measured. It is necessary to know the noise on the sensors so that the initial covariance matrix for the Kalman filter can be calculated. The noise on the magnetometer and sun sensor are shown in Figures 7.29 and 7.30.

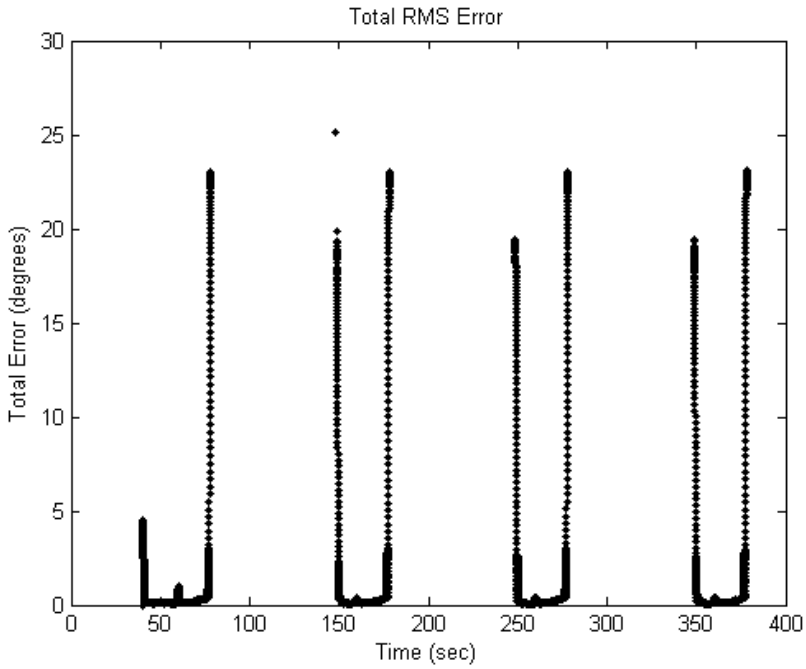

Fig. 7.27: Total RMS error in results found using low pass filter.

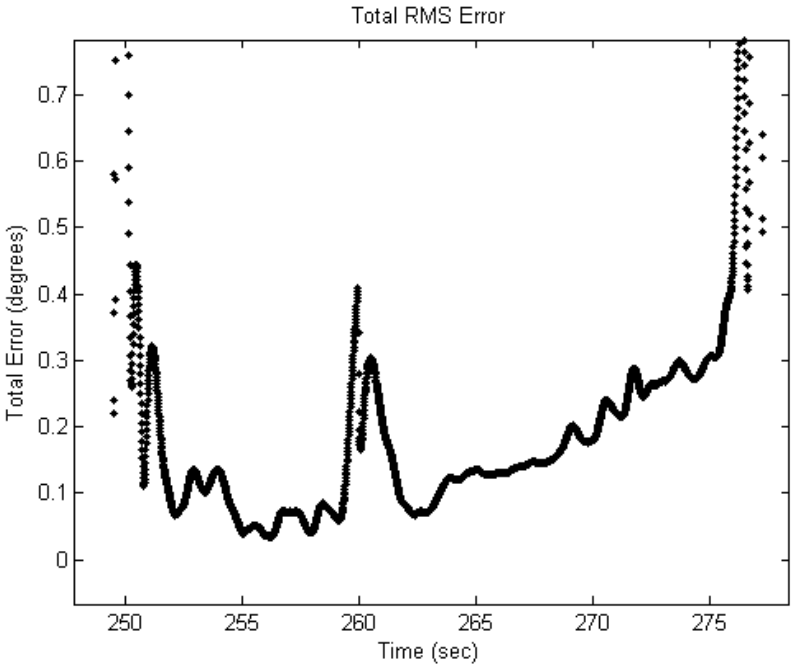

Fig. 7.28: Total RMS error in results found using low pass filter for one pass.

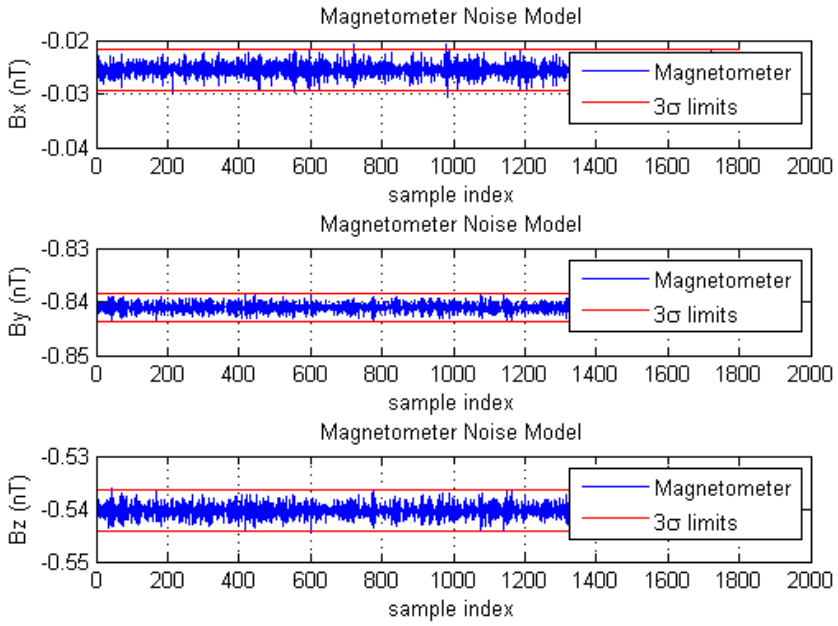

Fig. 7.29: Noise on magnetometer.

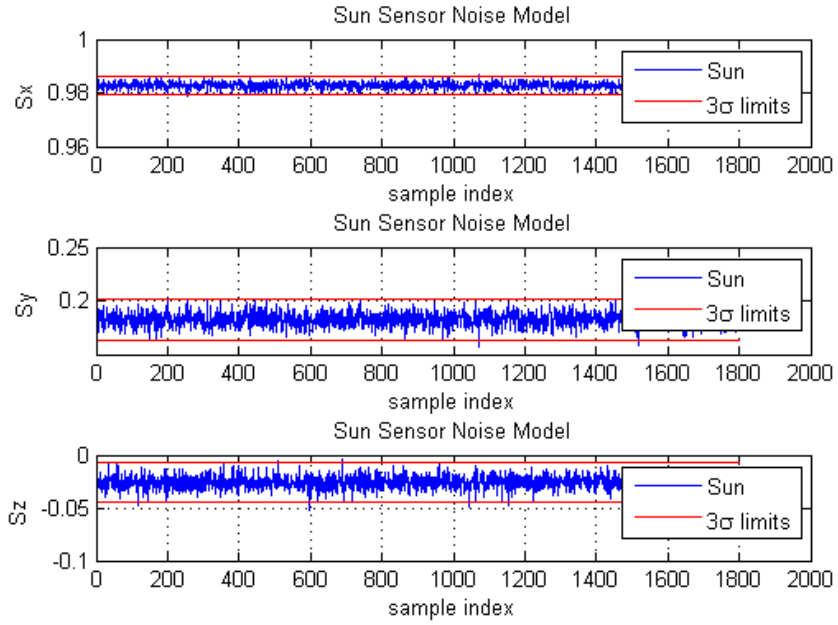

Fig. 7.30: Noise on sun sensor.

Table 7.3: Kalman Filter Results

Test	Kalman Filter Error with Sun (deg.)		Kalman Filter Error without Sun (deg.)	
	Error	Predict	Error	Predict
Spin 1	0.28	0.043	1	0.291
Spin 2	0.19	0.043	1.8	3
Spin 3	0.3	0.25	0.7	0.4

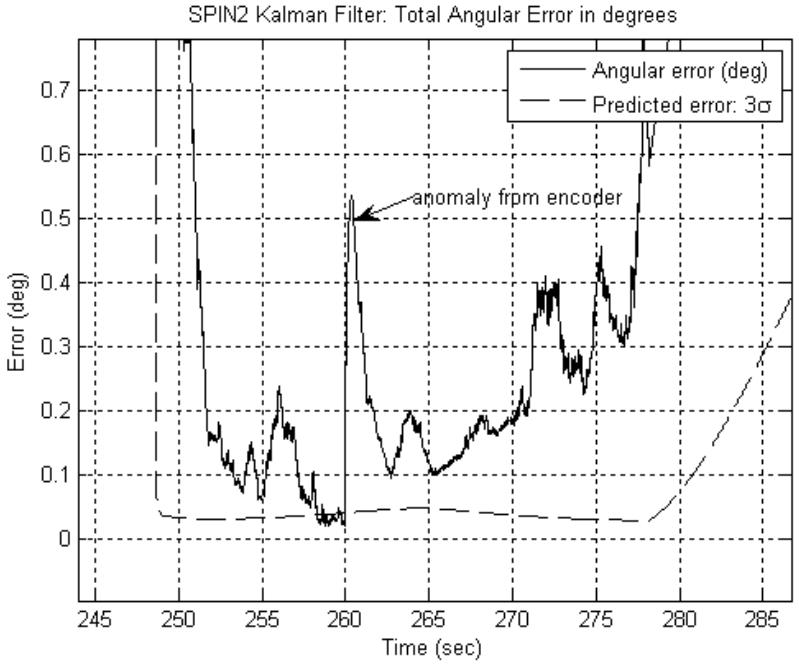

Fig. 7.31: Kalman filter error with sun for one pass.

Table 7.3 shows the results found using the Kalman filter. The error found using the Kalman filter is shown in Figures 7.31 and 7.32.

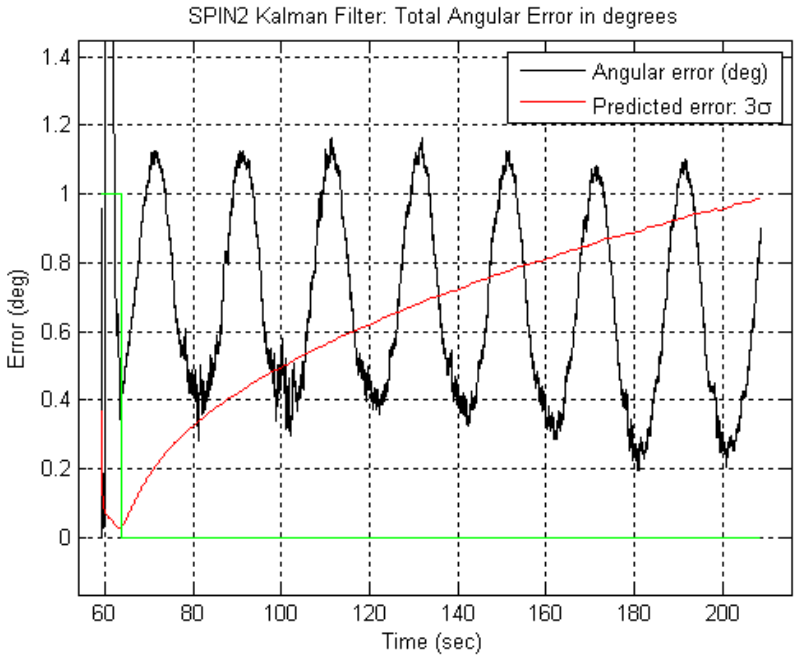

Fig. 7.32: Kalman filter error with no sun.

Chapter 8

Conclusions

The NOVA laboratory is capable of accurately creating sensor excitation sources for sun and magnetometer sensors. It is currently capable of being used to test and calibrate the DICE spacecraft. Attitude determination solutions can be accurately evaluated in the 0.1° to 1° error range. Calibration methods have been developed to estimate the sun and magnetometer sensor models, their mounting offsets, and the inertial vector to the sun.

Assigning a specific error value to a specific technique or measurement set does a disservice to the data. Deterministic errors tend to dominate the errors, especially as the sun sensor reaches the limits of its FOV.

Three-axis attitude errors for ranged from 0.25° to 0.60° when the sun is in the field of view for the mock satellite using the Dowty TAM and the SDL two-axis sun sensor. The attitude errors are dominated by deterministic effects, most probably due to calibration inaccuracies. Attempts to low pass filter the sensor data resulted in phase lag errors competitive in magnitude with that introduced by the noise. However random errors can be greatly reduced in post-processing by using a low pass smoother on the sensor data. As expected, the lowest errors result from the application of the Kalman filter algorithm, followed by the Q method and then the TRIAD method. However the TRIAD method was not significantly worse than the other methods, and its computational simplicity may make it the algorithm of choice for flight.

The size of the deterministic errors in the solution indicates that the calibration methods used are insufficient to evaluate attitude solution errors of under 0.5°. The calibration variations between tests should be investigated to determine whether these are due to time or temperature drift or simply inadequate calibration tests. If so, improved testing methods, analysis and sensor models need to be examined. One standard approach is to include the

calibration parameters as additional states in the Kalman filter. However this would increase the post-processing computational efforts. A full uncertainty analysis should be performed to address this problem.

Finally, the Kalman filter must propagate attitude solutions based on only magnetometer data for a large part of the mission. This has been shown to be successful where random noise dominates the error measurements . Deterministic errors cause a significant increase in these errors over those predicted from the Kalman filter. This is the principal reason to reduce these errors.

References

[1] Wertz, J. R., *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1978.

[2] Vallado, D. A. and McClain, W. D., *Fundamentals of Astrodynamics and Applications*, Vol. 21, Microcosm Press, El Segundo, CA, 2007.

[3] Sidi, M. J., *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, New York, NY, 1997.

[4] Wie, B., *Space Vehicle Dynamics And Control*, American Institute of Aernautics and Astronautics, Reston, VA, 2008.

[5] Bak, T., *Spacecraft Attitude Determination - A Magnetometer Approach*, Ph.D. thesis, Aalborg Universitetsforlag, 1999.

[6] Black, H. D., "A Passive System for Determining the Attitude of a Satellite," *AIAA Journal*, Vol. 2, No. 7, Jul. 1964, pp. 1350–1351.

[7] Shuster, M. D., "The Triad Algorithm as Maximum Likelihood Estimation," *The Journal of the Astronautical Sciences*, Vol. 54, No. 1, Jan. 2006, pp. 113–123.

[8] Wahba, G., "A Least Squares Estimate of Satellite Attitude," *SIAM Review*, Vol. 7, No. 3, Jul. 1965, pp. 409.

[9] Shuster, M. D. and Oh, S. D., "Attitude Determination from Vector Observations," *Journal of Guidance and Control*, Vol. 4, No. 1, Jan.-Feb. 1981, pp. 70–77.

[10] Cheng, Y. and Shuster, M. D., "Speed Testing of Attitude Estimation Algorithms," 2008, (to be published).

[11] Markley, F. L., "Attitude Determination Using Vector Observations and the Singular Value Decomposition," *The Journal of the Astronautical Sciences*, Vol. 36, No. 3, Jul.-Sept. 1988, pp. 245–258.

[12] Markley, F. L., "Attitude Determination Using Vector Observations: A Fast Optimal Matrix Algorithm," *The Journal of the Astronautical Sciences*, Vol. 41, No. 2, Apr.-Jun. 1993, pp. 261–280.

[13] Mortari, D., "ESOQ: A Closed-Form Solution to the Wahba Problem," *The Journal of the Astronautical Sciences*, Vol. 45, No. 2, Apr.-Jun. 1997, pp. 195–204.

[14] Mortari, D., "Second Estimator of the Optimal Quaternion," *Journal of Guidance, Control and Dynamics*, Vol. 23, No. 5, Sept.-Oct. 2000, pp. 885–888.

[15] Bruccoleri, C., Lee, D.-J., and Mortari, D., "Single-Point Optimal Attitude Determination Using Modified Rodrigues Parameters," *The Malcolm D. Shuster Astronautics Symposium*, Buffalo, NY, 2005.

[16] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, No. Series D, 1960, pp. 35–45.

[17] E. J. Lefferts, F. L. M. and Shuster, M. D., "Klaman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 5, No. 5, Sept.-Oct. 1983, pp. 417–429.

[18] Jandak, M. and Fullmer, R. R., "An Extedned Kalman Smoother for the DICE Mission Attitude Determination Post Processing with Double Field Probe Inclusion," *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, AK, 2011.

[19] Maybeck, P. S., *Stochastic Models, Estimation, and Control*, Vol. 1, Academic Press, New York, NY, 1979.

[20] Chobotov, V. A., *Spacecraft Attitude Dynamics and Control*, Krieger Publishing Company, Malabar, FL, 1991.

[21] Elsasser, W. M., "Hydromagnetic Dynamo Theory," *Reviews of Modern Physics*, Vol. 28, No. 2, Apr. 1956, pp. 135–163.

[22] Fabiano, E. B. and Peddie, N. W., "Grid Values of Total Magnetic Intensity IGRF - 1965," *ESSA Technical Report C&G*, Vol. 38, Apr. 1969, pp. 55.

[23] URL:http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html.

[24] URL:http://www.ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml.

[25] Chetty, P., *Satellite Technology and Its Applications*, TAB Books, Blue Ridge Summit, PA, 1991.

[26] Fortescue, P., Stark, J., and Swinerd, G., editors, *Spacecraft Systems Engineering*, John Wiley and Sons Ltd., West Sussex, England, 2003.

[27] Sripruetkiat, P., *The Use of Complementary Metal-Oxide-Semiconductor (CMOS) Camera Data for Spacecraft Attitude Determination*, Master's thesis, Utah State University, 2006.

[28] Mortari, D. and Gigli, S., "Earth-Sun Attitude Sensor: Hardware Design and Ground Tests," *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, AK, 1999.

[29] Chang, T., *Advances in Spacecraft Technologies*, InTech, Rijeka, Croatia, Feb. 2011.

[30] Poppenk, F. M., Amini, R., and Brouwer, G. F., "Design and Application of a Helmholts Cage for Testing Nano-Satellites," *6th International Symposium on Environmental Testing for Space Programmes*, 2007.

[31] Craig, J. J., *Introduction to Robotics*, Addison-Wesley, Reading, MA, 1989.

[32] Ryan, K. P., Fullmer, R. R., and Wassom, S. R., "Experimental Testing of the Accuracy of Attitude Determination Solutions for a Spin-Stabilized Spacecraft," *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, AK, 2011.

Appendix

Sun Sensor Calibration Coefficients

Table A.1: Sun Sensor Calibration Coefficients

Coefficients	$X1^5$	$X2^5$	$Y1^5$
Scaled-Y - 1st Order	0	0	0
Scaled-Z - 1st Order	0	0	0
Scaled-Y - 2nd Order	0	0	0
Scaled-Z - 2nd Order	0	0	0
Scaled-Y - 3rd Order	0	0	0
Scaled-Z - 3rd Order	0	0	0
Scaled-Y - 4th Order	0	0	0
Scaled-Z - 4th Order	0	0	0
Scaled-Y - 5th Order	-27.50259846	65.65148333	17.36656123
Scaled-Z - 5th Order	28.93423584	-14.75248093	-31.9366148

Table A.2: Sun Sensor Calibration Coefficients Continued

$Y2^5$	T^5	$X2^4$	$X1^4$	$Y1^4$
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	12.08995852	-12.4487067	14.34988951
0	0	2.914074823	-14.69839205	-7.124650627
-28.60135195	-3962.671807	-94.00376138	32.52849519	-12.92053148
14.77493049	-6107.754576	20.04022226	-69.40962197	50.75677229

Table A.3: Sun Sensor Calibration Coefficients Continued

$Y2^4$	T^4	$X1^3$	$X2^3$
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	4.517310827	-4.001573542
0	0	6.906735075	-7.197336949
-4.380931222	337.9217991	20.71085279	-19.11889721
14.46677886	-9.313706079	27.16954213	-9.03900659
43.67504824	18295.25937	-6.355016111	47.0264629
-4.508229575	27794.92838	67.16930836	-14.18875521

Table A.4: Sun Sensor Calibration Coefficients Continued

$Y1^3$	$Y2^3$	T^3	$X1^2$
0	0	0	0
0	0	0	0
0	0	0	14.62931223
0	0	0	4.622899677
-3.780676976	2.8717866	-6.920104602	1.77867533
4.944605272	-3.826701695	14.6988433	-13.87963175
-24.13920839	8.253944108	-1252.83469	-4.240746246
14.23451021	-22.62534156	51.21047931	-25.78666316
-8.590299624	-22.74523932	-33787.31449	3.545123197
-26.48562912	-15.4340325	-50557.68404	-39.67394612

Table A.5: Sun Sensor Calibration Coefficients Continued

$X2^2$	$Y1^2$	$Y2^2$	T^2
0	0	0	0
0	0	0	0
13.03325767	-13.88853397	-12.79394861	-3.697379276
3.596561658	-3.804169865	-5.21800708	4.882259109
9.913719217	-2.156625046	-8.641526797	14.54671505
0.880636742	0.517986531	9.296208929	-39.66439608
19.09320322	6.600932499	-13.50633415	1739.073292
-0.702771216	-2.488693914	20.18456588	-95.12875617
-0.602212207	2.383133786	-4.103197996	31197.08169
-0.918626971	11.17268839	19.57740863	45945.29177

Table A.6: Sun Sensor Calibration Coefficients Continued

$X1$	$X2$	$Y1$	$Y2$	T
1.822317471	-2.208448242	-2.034987992	2.001531399	0.077101327
1.568999468	-2.436432123	1.677722328	-2.315521996	0.312015878
-14.35162098	-17.42481659	8.442427613	11.93812867	6.452765301
1.821117813	-1.653959378	9.945780274	6.821432638	-9.114425631
-1.040046041	-7.364377458	4.191341117	10.10054376	-9.189683174
-0.609344757	-9.334006761	-12.80167082	-19.31793193	35.57675875
244.4599659	234.9295789	249.2312525	258.2274556	-1071.192687
-6.933864044	-17.75424812	-23.6559799	-33.13981116	73.91024605
2654.725222	2649.076955	2661.301096	2668.406287	-14400.88666
3801.094998	3788.272923	3779.967109	3772.519492	-20859.91604

Detailed Test Procedure

A.0.1 Power up Sun Source (Laser)

When the sun sensor was calibrated, the laser warm up for at least four hours. It is best to let the laser stay on overnight. When the laser is warm it apparently fluctuates less which yields more accurate measurements and in turn gives better results. A Panavise is used to align the laser so that it shines directly on the aperture on the sun sensor. While the laser is warming up it is best to lay it on the work bench with the laser beam covered to prevent people from looking at the beam.

A.0.2 Matlab XPC Target

The PC-104 runs Matlab XPC Target using the embedded stand alone mode. XPC Target requires the target computer to always be connected to the host computer in order to run. Because the PC-104 will be spinning on the rotation table it is necessary to run the PC-104 in the embedded stand alone mode.

If it is desired to repeat the test that the author conducted skip to Subsection A.0.3. If it is desired to run a different model, than follow the procedure explained in this Section.

Turn on Embedded Stand Alone

To turn on the embedded stand alone mode the model that is designed needs to have the Configuration Parameters changed. To change the parameters go to the ***Simulation*** tab above the model and select ***Configuration Parameters***. In the ***Configuration Parameters*** go to the ***Real-Time Workshop*** tab that is on the left and change ***System target file*** to ***xpctargetert.tlc***. This will enable the embedded stand alone mode. Now when a model is built the files that are needed to run the model in stand alone mode will be created. After the files are created, it is then necessary to load these files onto the PC-104.

Load Model

1. Connect the keyboard, monitor, and laptop (using an Ethernet cable) to the PC-104.
2. Connect the power supply to the PC-104 (so as not to waste the batteries) and turn on the PC-104. The PC-104 should boot to the xpctarget screen and wait for your next command. If the PC-104 does something else, refer to Subsection A.0.4 and follow steps 25 through 29, but skip 27, to get the PC-104 to boot to the standard XPC Target mode.
3. Open Matlab on the laptop.
4. On the laptop open ***xpc target explorer*** by executing the command ***xpcexplr*** in the Matlab command window.
5. In the ***xpc target explorer*** window, right click on ***Target PC***, select the ***Connect***.
6. In the ***xpc target explorer*** window under the ***Target PC,*** select the ***Configuration***.
7. In the ***Configuration*** window change the ***Target boot mode*** to ***Stand Alone***. This will allow the PC-104 to run in the embedded stand alone mode.
8. Build the Simulink model that needs to be ran by the PC-104. Matlab will create some folders and files in the working directory.
9. Copy and paste the ***ftp_put.m*** and ***get_data.m*** scripts into the ***.emb*** folder that was created in the previous step.
10. In ***ftp_put.m*** change the ***.rtb*** to the name of the new ***.rtb*** file that was created in step 8.
11. Send the PC-104 the files created in step 8 by running ***ftp_put.m***. Make sure that the ftp_put.m is in the .emb folder so that the files can be sent.
12. Turn off the PC-104 and disconnect Ethernet, keyboard, monitor, and power. Make sure that the PC-104 can rotate freely. The PC-104 will automatically run the model that was just put on it when it is turned on next.

The embedded option will overwrite previous data sets. If data is collected, that data has to be retrieved before any more data can be collected. If the model is ran again after data is collected, all the previous data will be deleted. How data is retrieved is explained in the Subsection A.0.4.

A.0.3 Set up Hardware

After the white stand and PC-104 are attached to the rotation table, align the inertial coordinates with the inertial magnetometers coordinates rotated 180° about the Y axis.

A.0.4 Data Collection

Temperature

As explained in the Chapter 3, the PC-104 has one less differential analog input than desired and so the temperature needs to be replaced by a constant. It is still necessary to know what the output is for the temperature so that an accurate constant can be used. For this reason, the first and last tests will collect data with the temperature from the sun sensor connected, for all the tests in-between the sun sensor temperature will not be connected. The first and last tests were chosen because it was believed that the temperature output would not change much. If the first and last test show there was really no change in the temperature, than it can be assumed that there was no change in temperature during the rest of the tests.

For the first and last test it is not necessary for the mock satellite to spin, and so the PC-104 can just be plugged into the power supply so that the batteries can be saved for other tests. Also, since the mock satellite is not spinning the PC-104 can run in the standard XPC Target mode. This means that the the monitor, keyboard, and laptop have to all remain connected to the PC-104, this will save time by not having to disconnect and connect the cables. To use XPC Target in the standard mode a new model will be needed.

For the test the inertial computer does not need to collect any data. Although the only measurement needed is the temperature from the sun sensor the remaining seven analog in-

puts might as well record data. This makes it easier to be sure that everything is functioning by being able to see the magnetometer and sun sensor outputs. When the temperature data was being collected the light sensor was left disconnected.

Procedure

1. Put the laser in the Panavise and align it so that the beam is shinning directly on the aperture of the sun sensor.
2. Disconnect light sensor from the DAQ card #4 slot.
3. Connect sun sensor temperature to DAQ card #4 slot.
4. Connect the power, keyboard, monitor, and laptop (using an Ethernet cable) to the PC-104.
5. Turn on the PC-104, the switch for the PC-104 is the switch that is on the right when looking at the three red switches on the side of the power supply.
6. The PC-104 should boot to the standard XPC Target screen. If this does not occur follow steps 25 through 29 and skip 27 in Subsection A.0.4.
7. Build, connect, and run the new Simulink model that will run in the standard XPC Target mode.
8. Save the data. The data from the sensors is stored in an object and is in ***tg.OutputLog***, the time is in the same object and can be found in ***tg.TimeLog***. The can then be saved as arrays and then saved to a flash drive.

Spin Test and Sweep Test

The procedure for the two different tests are very close to the same, the only thing that needs to be changed is some code in the controller for the motor. The controller for the motor is found in the same Simulink model as the controller for the Helmholtz cage. In the controller there is a "User Defined Function" called ***fcn***. There are several constants

that are inputs to ***fcn***. ***Angular Vel.*** is the constant that sets the angular velocity of the motor. ***Offset*** is the time offset between when the Simulink model starts to run and the motor starts to spin. Inside of ***fcn*** there is a comment that says "Spin Test" and about ten lines below that there is a comment that says "Sweep Test". If it is desired to conduct a spin test then comment out the remaining code below the "Sweep Test" comment. If it is desired to conduct a sweep test then comment out all the code in-between the "Spin Test" and "Sweep Test" comments.

Procedure

If the Simulink model was just loaded skip to step 6.

1. Connect the power supply, keyboard, monitor, and laptop (using an Ethernet cable) to the PC-104.

2. Turn on PC-104. While it is booting, press ***F5*** when prompted to stop it from booting and to be brought to the DOS command line. There will only be a window of a couple of seconds to press F5, as soon as the message appears to press F5, it needs to be pressed, or the computer will continue booting.

3. Type and execute ***cd xpcfiles*** to change to the xpcfiles directory.

4. Type and execute ***dir*** to see the different files in the directory.

5. Type and execute ***copy autoexec.sm1 autoexec.bat***, this command overwrites autoexec.bat with the file needed to run the model. The file autoexec.sm1 was created by Matlab when the desired model was built. The autoexec.sm1 was saved by completing step 28 at a previous time. If a different model is desired to be ran, replace sm1 with the three letters or number of the desired model.

6. Turn off the PC-104 and disconnect the power supply, keyboard, monitor, and laptop from the PC-104. The PC-104 will automatically start collecting data the next time the PC-104 turns on.

7. Make sure the mock satellite can rotate freely, that the surface is still level that the motor is on, and make sure the laser beam is shining directly on the aperture of the sun sensor.

8. In the inertial computer, set the motor to spin or sweep and at the desired rate.

9. In the inertial computer change the magnetic field to what is desired for the current test.

10. Build the controller on the inertial computer and initialize it after it is built.

11. Make sure the motor has power. When looking at the red switches on the power supply, the middle switch is for the rotation table.

12. Make sure the motor is connected to the inertial DAQ card correctly.

13. Make sure the inertial magnetometer is connected to the inertial DAQ card correctly.

14. Make sure the inertial light sensor is connected to the inertial DAQ correctly. Also make sure the inertial light sensor's power supply is on and is providing five V to the light sensor.

15. Turn on the power supplies, relays, and magnetometer for the Helmholtz cage.

16. Turn on the strobe so that it flashes approximately four times a second. The strobe cannot be set to an exact frequency, the dial on the back just allows one to make it flash faster or slower.

17. Make sure that in the motor controller the delay for the motor is sixty seconds. Sixty seconds are needed to plug in the PC-104, manually home the motor, and to give the rotation table time to get close to home.

18. Begin running the inertial controller, as soon as the inertial controller starts to run, put on a wrist strap and step inside the Helmholtz cage and plug in the battery for the PC-104, it will take approximately thirty seconds for the PC-104 to boot up.

19. As soon as the battery is plugged in, manually turn the rotation table through home. Make sure to not move the optical bread board that the rotation table is on. Get out of the cage.

20. Turn off the lights. It is easier for the light sensors to see the flashes from the strobe with the lights off.

21. Let test run for the desired amount of time.

22. While the test is running, turn on and off and back on the lights. This will make it easier to align the time vectors. It is a lot easier to align when the lights go off than on because it takes more time for the lights to come on.

23. Stop the Simulink model on the inertial computer and disconnect the battery to the PC-104.

24. Turn off the strobe.

25. Reconnect the laptop, keyboard, monitor, and power supply to the PC-104.

26. Turn on the PC-104. While PC-104 is booting press ***F5*** when prompted to go to the DOS command line. If F5 is not pressed in time the PC-104 will continue booting and start collecting another set of data. The data that had just been collected will be erased if F5 is not pressed in time.

27. On the DOS command line type and execute ***cd xpcfiles*** to change to the xpcfiles directory. Type and execute ***dir*** to see the files in the directory.

28. If this is the first time that the PC-104 has ran this model it would be beneficial to save the model as a different name. If this model has been ran before it is not necessary to save it again and this step can be skipped. Type in the command line ***copy autoexec.bat autoexec.???*** and execute it. Replace the ??? with any three letters or numbers. This is necessary to do because autoexec.bat will be overwritten by a different file and saving the current autoexec.bat to a different name will make it so the file won't have to be loaded again.

29. Type and execute ***copy autoexec.xpc autoexec.bat*** in the command line. The user will be asked if they desire to overwrite the current autoexec.bat, type ***y*** to overwrite.

30. Boot the PC-104 in the standard XPC Target mode by typing and executing ***autoexec.bat*** on the command line.

31. On the laptop in the xpc target explorer window right click on the Target PC and click on connect to connect to the PC-104.

32. Get the data that was collected by the PC-104 by running ***get_ data.m***.

33. Save the data from PC-104. The data is retrieved from the PC-104 as eight different structures. The data can be saved to a flash drive as structures or as arrays.

34. Save the data from the inertial computer. The data from the inertial computer is in an object. So that the data can be saved it needs to be saved to arrays. The data from the sensors is in ***tg.OutputLog*** and the time is in ***tg.TimeLog***.

35. Now that the data is saved another test can be ran. Step 27 can be skipped for each additional test.

Derivation of Q Method

The derivation of the q method is shown in Equation A.1 through EquationA.11. The derivation starts out with the Wahba problem.

$$
\begin{aligned}
J(\hat{A}) &= \frac{1}{2}\sum_{j=1}^{N}\sigma_j^{-2}[\tilde{b}_j - \hat{R}r_j]^T[\tilde{b}_j - \hat{R}r_j] && \text{(A.1)}\\
&= \frac{1}{2}\sum_{j=1}^{N}\sigma_j^{-2}[\tilde{b}_j^T\tilde{b}_j - (\hat{R}r_j)^T\tilde{b}_j - \tilde{b}_j^T(\hat{R}r_j) + r_j^T\hat{R}^T\hat{R}r_j] && \text{(A.2)}\\
&= -\sum_{j=1}^{N}\sigma_j^{-2}\tilde{b}_j^T(\hat{R}r_j) + \frac{1}{2}\sum_{j=1}^{N}\sigma_j^{-2}[\tilde{b}_j^T\tilde{b}_j + r_j^T r] && \text{(A.3)}\\
&= -\sum_{j=1}^{N}\sigma_j^{-2}\tilde{b}_j^T(\hat{R}r_j) + const. && \text{(A.4)}
\end{aligned}
$$

If $\hat{R}$is replaced with $\Xi^T(q)\Psi(q)$ (q stands for the quaternion), and if it is noted that the constant doesn't effect the minimum then Equation A.4 can be expressed as Equation A.5.

$$J(q) = -\sum_{j=1}^{N}\sigma_j^{-2}\tilde{b}_j^T\Xi^T(q)\Psi(q)r_j \qquad \text{(A.5)}$$

Using quaternion relations it can be proved that $\Xi^T(q)w = \Omega(w)q$ and $\Psi(q)w = \Gamma(w)q$. With these new equalities Equation A.5 can be expressed as Equation A.6.

$$
\begin{aligned}
J(q) &= -\sum_{j=1}^{N}\sigma_j^{-2}q^T\Omega^T(\tilde{b}_j)\Gamma(r_j)q && \text{(A.6)}\\
&= -q^T(\sum_{j=1}^{N}\sigma_j^{-2}\Omega^T(\tilde{b}_j)\Gamma(r_j))q && \text{(A.7)}\\
&= -q^TKq && \text{(A.8)}
\end{aligned}
$$

To find the "best R" the problem is redefined to find $\hat{q}$ which maximizes $J(q) = q^TKq$ and that $\hat{q}^T\hat{q} = 1$. Using Lagrange Multipliers, an additional constraint of $J(q) = q^TKq +$

$\lambda(1 - q^T q)$ can be added. Equation A.8 can then be as Equation A.9.

$$\frac{\partial J(q)}{\partial q} = 2Kq - 2\lambda q|_{q=\hat{q}} = 0 \tag{A.9}$$

$$K\hat{q} = \lambda\hat{q} \tag{A.10}$$

When Equation A.10 is substituted back into J the result is Equation A.11.

$$J(q) = \hat{q}K\hat{q} = \hat{q}\lambda\hat{q} = \lambda \tag{A.11}$$

The quaternion that maximizes J is the normalized eigenvector that corresponds to the largest eigenvalue for the matrix K. As shown in Equation A.8 $K = \sum_{j=1}^{N} \sigma_j^{-2} \Omega^T(\tilde{b}_j)\Gamma(r_j)$. An optimal quaternion will always be able to be found if there are two observation vectors that are not parallel.